

## PROJECT ADMINISTRATION DATA SHEET



ORIGINAL



REVISION NO. \_\_\_\_\_

Project No. E-24-655DATE 4/20/82Project Director: Dr. W. B. RouseSchool/Lab ISyESponsor: US Defense Supply Service - WashingtonType Agreement: Contract No. MDA903-82-C-0145Award Period: From 6/1/82 To 6/1/86 (Performance) 6/1/85 (Reports)Sponsor Amount: \$427,464 (\$133,091 partially funded thru 9/30/82)\* Contracted through:

Cost Sharing: \_\_\_\_\_ GTRI/STP

Title: Human Problem Solving in Complex Dynamic Environments

## ADMINISTRATIVE DATA

OCA Contact Linda H. Bowman x4820

## 1) Sponsor Technical Contact:

Dr. J. CRASANK5001 Eisenhower Ave.Alexandria, VA 22333202-274-8840

## 2) Sponsor Admin/Contractual Matters:

Mr. Thomas A. BryantONRRRGIT214 O'Keefe BldgAtlanta, GA 30332404-881-4213Defense Priority Rating: noneSecurity Classification: none

## RESTRICTIONS

See Attached gov't Supplemental Information Sheet for Additional Requirements.

Travel: Foreign travel must have prior approval - Contact OCA in each case. Domestic travel requires sponsor approval where total will exceed greater of \$500 or 125% of approved proposal budget category.

Equipment: Title vests with sponsor; under \$1,000 vests with GIT if prior sponsor approval obtained.

## COMMENTS:

\* 9/30/82 is merely an estimate, appropriated funds can be spend beyond that date.

## COPIES TO:

Administrative Coordinator  
Research Property Management  
Accounting  
Procurement/EES Supply Services  
FORM OCA 4:781Research Security Services  
Reports Coordinator (OCA) ✓  
Legal Services (OCA)  
LibraryEES Public Relations (2)  
Computer Input  
Project File  
Other \_\_\_\_\_

SPONSORED PROJECT TERMINATION/CLOSEOUT SHEETDate 4-24-87Project No. E-24-655School/Dept XXB ISyEIncludes Subproject No.(s) N/AProject Director(s) W.B. RouseGTRC XXBSponsor Defense Supply Service - WashingtonTitle Human Problem Solving in Complex Dynamic EnvironmentsEffective Completion Date: 6/1/86 (Performance) 6/1/86 (Reports)

## Grant/Contract Closeout Actions Remaining:

☐

None

☒

Final Invoice or Final Fiscal Report

☒

Closing Documents

☒

Final Report of Inventions - Questionnaire sent to P.I.

☒

Govt. Property Inventory &amp; Related Certificate

☐

Classified Material Certificate

☐

Other \_\_\_\_\_

Continues Project No. \_\_\_\_\_ Continued by Project No. \_\_\_\_\_

## COPIES TO:

Project Director  
Research Administrative Network  
Research Property Management  
Accounting  
Procurement/GTRI Supply Services  
Research Security Services  
~~Reports Coordinator (OCA)~~  
~~Legal Services~~

Library  
GTRC  
~~Research Communications (2)~~  
Project File  
Other Diane H.  
Angela DuBose  
Russ Embry

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Bimonthly Report

For the Period June 1, 1982 - July 31, 1982

For

Contract MDA 903-82-C-0145  
(June 1, 1982 - July 1, 1982)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404-894-3996)

This program of research was initiated during this reporting period. Most of our efforts during this period have been devoted to the design and programming of a problem solving scenario involving the hierarchial display of a multi-page representation of a dynamic message processing network. A prototype system should soon be ready for testing.

Also during this period, efforts were invested in collecting literature on alternative approaches to displaying information about large-scale systems. As expected, very few empirical results were located, but several interesting although untested design concepts were found.

As of July 31, 1982, approximately \$9,000 had been spent.

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Bimonthly Report

For the Period August 1, 1982 - September 30, 1982

Contract MDA 903-82-C-0145  
(June 1, 1982 - July 1, 1983)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404/894-3996)

During this reporting period, work continued on the problem solving scenario which is now termed MABEL (Monitoring, Assessing, Browsing, and Exploring Limits.) An initial version of this hierachical display of a multi-page representation of a dynamic message processing network is now operational and an experiment will soon be underway involving three independent variables: 1) number of elements per page, 2) number of levels of pages, and 3) message load scenario. The attached paper summarizes MABEL and the experimental plans; however, as this paper was written one month ago, it is already a little out of date with regard to the format and operations (and even the name) of MABEL.

As of September 30, 1982, approximately \$12,000 had been spent.

HUMAN PROBLEM SOLVING  
IN LARGE SCALE NETWORKS

Richard L. Henneman and William B. Rouse

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, Georgia 30332 USA

ABSTRACT

The role of the human operator in monitoring and control of large scale networks is discussed. An experimental scenario is presented that has been developed from the study of factors affecting humans in this role. An experiment currently in progress is considered and reasearch goals discussed.

INTRODUCTION

Current trends in computer technology are leading to the development of many highly integrated systems in the domains of communications, transportation, manufacturing, etc. Most of these systems can be represented as large networks of nodes and arcs where nodes denote people, destinations, or machines and arcs denote communication lines, transportation routes, or a variety of activities. It is not unusual for there to be hundreds or thousands of nodes and arcs. Networks of this size and level of interconnectivity are truly complex systems.

The complexity of problems in general, and of these networks in particular, is also affected by their dynamic nature [1]. The states of the nodes and arcs (i.e., levels, flows, etc.) usually evolve in time and are not amenable to instantaneous control. Further, the demands placed upon the networks are often time-varying, with occurrences of peak demanda not always being predictable.

There are two primary ways in which humans become involved with complex dynamic networks. The first, and probably the most familiar, way is as users. Most people are users of communications and transportation networks. The use of command and control networks by military

commanders has received considerable attention [2,3,4,5,6]. The problem addressed in this paper, however, is not the user but instead is the network controller or operator.

The task of the network controller is to manage the assets of the network (i.e., nodes and arcs) so as to maximize network efficiency [7]. Further, during peak demand periods, the controller may have to implement control procedures such as load shedding and priority scheduling to assure that overloads do not degrade network performance [8].

For many aspects of this task, the network controller has computer aids or, in fact, may simply have to monitor an automatic system which performs many functions [9,10]. However, system failures or unusual environmental demands can require that the human intervene and manually control the network. The human's abilities to solve these types of problem are not well understood.

Several issues need to be considered. The first issue is the human's ability to decide that intervention is necessary. While increasingly sophisticated alarm techniques may handle most of this function [11], human factors research is still needed to determine the best way to display sophisticated alarm information.

Given that intervention is necessary, the human must then coordinate two goals: compensation and diagnosis. Compensation involves configuring the network so as to counteract the symptoms of the problem and continue network operation. Diagnosis involves collecting information so as to isolate the source of the problem. The coordination of these two goals is complicated by the fact that delaying diagnosis may result in total loss of the network's functions. On the other hand, focusing on diagnosis may result in an unacceptable interruption of network operation.

While considerable attention has been devoted to the human's diagnostic abilities [12], only a limited amount of research has dealt with the human's abilities to coordinate compensation and diagnosis [13,14,15]. Further, little if any research has focused on truly large dynamic networks. The program of research upon which this paper is based is focusing on the tasks of compensating for and diagnosing problems in complex dynamic networks.

In the following section, the general experimental scenario being used for this research program is discussed. Then, an experimental investigation currently in progress is considered. Finally, the longer-term goals of this research program are briefly summarized.

#### EXPERIMENTAL SCENARIO

LCDS (Large Complex Dynamic System) is a simulation of a large-scale queueing network. It is programmed in Pascal on a VAX 11/780, and operates in real time. The network is not designed to represent any one type of large-scale dynamic network; instead, it is designed to simulate features of various complex systems. In general, LCDS is structured as groups of networks which, in turn, are structured as larger networks. Below is a summary of LCDS, beginning with a description of its basic structural elements, followed by a description of how the system typically operates in both normal and failure situations.

##### Network Structure

Several elements are basic to the structure of LCDS. A node represents the smallest structural unit in the network. Customers are passed from node to node, following a path that will minimize their expected time in the system. Thus, customers form queues at nodes, waiting until they can be serviced and passed on to the next nodes in their paths. Nodes are grouped into network structures called clusters. The cluster structure defines the most elemental relationship between nodes. Clusters are grouped accordingly into levels in the hierarchy. Levels define the relationships between clusters. Thus, customers travel horizontally through the system via nodes which are contained within clusters. Customers travel vertically through the system via clusters, which are contained within levels.

Typically a customer enters the system through a node which is contained in a cluster at the lowest level. The customer must work its way horizontally

through the system until it reaches the end of its current cluster. At this point, the customer is transferred to the next level through a connecting cluster. In this way, the customer moves up or down through the system from its source to its sink node.

The system can perhaps be best visualized by imagining a pyramid. Each block in the pyramid's structure is supported by a group of stones underneath it. Similarly, in LCDS, every node in a cluster is "supported" by another group of clusters in a lower level.

Customers arrive at the lowest level in the system according to a Poisson process. Service times are distributed exponentially with a constant mean for all nodes within a given level. Upon arrival, customers are assigned both a source and sink node. The shortest horizontal path is determined for their current cluster, as well as their overall vertical path between levels.

##### Operator Interface

Important information about the system is presented to the network controller through a split-screen display format as shown in Figure 1. The right side of the screen displays a cluster of nodes, each of which is represented by a small rectangle. Due to the large number of nodes, the person monitoring LCDS is unable to ever view the entire network. Thus, only one cluster may be viewed on the screen at a time. By selecting the appropriate command (Figure 1), the user may travel up or down through the hierarchy, thereby viewing another cluster. Each node on the screen contains the current number of customers waiting to be serviced at that node. (In the cluster depicted, there are currently no customers waiting to be serviced.) Additionally, an indication on the side of the node represents the level of that node in the hierarchy.

The left side of the screen is used to display a variety of user-requested information. This information ranges from overall system performance statistics (total serviced, number in system, and average overall time in system) to detailed information about a particular node. The top of the screen shows the current time, while the bottom line prompts the user for his next command.

LCDS will, under normal conditions, control the path of customers efficiently without any interference from the operator. However, even though when a node fails the system will eventually avoid routing customers through it as the



Time = 6.56		1	5	9	13
<u>Command</u>	<u>Description</u>				
d**	Displays cluster beneath node ** in current display				
l**	Specifies **% of current customer demand to serve	2	6	10	14
n**	Gives specific information about node **				
r**	Checks node ** for failure, repairs if necessary				
s	Gives summary statistics for entire system				
u	Displays cluster above current cluster	3	7	11	15
h	Help: displays this list				
		4	8	12	16
Your Action: h					

Figure 1: LCDS Display

node's average service time increases, the system is unable to automatically diagnose failures. Hence, when a failure occurs, the human operator is confronted with a situation the system cannot handle on its own. Nodes can fail in two ways: 1) the node can simply break down and be unable to service customers; or 2) the number of customers in the node can exceed its maximum allowable queue size, thereby inducing a failure. It is easy to see that these failures will propagate through the system as nodes reach their maximum queue size and refuse to accept more customers.

#### Problem Solving Scenarios

The problems for the operator, therefore, become compounded as time proceeds. Not only must he detect and diagnose the failed nodes, but he must compensate for failures (in order to avoid further failures) by any of several different means: repairing nodes, reconfiguring the network, shedding load, or prioritizing customers.

An important consideration in LCDS, thus, is the time varying nature of the

number of customers requiring service in the network--the arrival rate. Large scale systems of this nature typically are busier at certain times than at others; hopefully, though, the system is designed such that it is capable of handling all of the load. However, during failures and certain high-level loading situations, the system is not capable of processing all of the load. The operator, thus, must recognize the overload situation and compensate for it by specifying the amount of load he wishes the system to service. Different problem solving scenarios are achieved in LCDS by varying the arrival rate as a function of time.

To summarize, subjects serving as operators in LCDS are required to monitor the hierarchical network for failures, compensating for both failures and the fluctuating demand.

#### EXPERIMENT IN PROGRESS

An experiment currently in progress has been designed to explore the impact of the structure of the large-scale hierarchy on the ability of the human to detect, diagnose, and compensate for failures.

The major independent variables in the study are the number of nodes displayed to the user on the screen (cluster size), number of levels, and arrival rate variations (problem solving scenario). The levels of the variables were selected to produce approximately the same number of nodes at the bottom level of each network. Thus, subjects monitor networks with: 1) 8 nodes/cluster and 4 levels; 2) 16 nodes/cluster and 3 levels; 3) 10 nodes/cluster and 4 levels; and 4) 20 nodes/cluster and 3 levels. (The 8 and 16 node/cluster networks result in 4096 nodes at the bottom level, while the 10 node/cluster and the 20 node/cluster networks result in 10000 and 8000 nodes, respectively, at the bottom level). The problem solving scenario is varied by having subjects monitor networks with 1) a constant arrival rate, and 2) an arrival rate that is initially constant, increases to some peak level, and decreases to the initial level). Subjects (who are graduate students in engineering) are required to monitor networks that vary in terms of the above factors, trying to serve as many customers as possible while both minimizing their waiting time in the system and coping with failures.

Major response variables include the average waiting time for customers, total number of customers served in a given period of time, average time to locate a failure, and total number of failures found by the operator. Also of interest are the search strategies used by subjects.

#### CONCLUSIONS

This brief paper has considered the role of the human in monitoring and control of large scale networks, and presented an experimental scenario developed to evaluate the factors that may affect this role. While the immediate goal of this research is to assess the effects of these factors, the longer-term goals include modeling human behavior in these types of tasks and developing aids to assist the human in such tasks. At this point in time, the model of human problem solving proposed by one of the authors [12] is being considered as a possible description of human behavior in the task outlined in this paper.

#### ACKNOWLEDGEMENT

This research is supported by the Army Research Institute for the Behavioral and Social Sciences under Contract No. MDA 903-82-C-0145.

#### REFERENCES

1. Hormann, A.M. 1971. "A Man-Machine Synergistic Approach to Planning and Creative Problem Solving," International Journal of Man-Machine Studies, Vol. 3, No. 2, pp. 167-184.
2. Baker, J. D. 1975. "SIMTOS: A Man-in-the-Loop Interactive Simulation of a Tactical Operations System," in R. K. Huber, L. F. Jones, and E. Reine, (Eds.), Military Strategy and Tactics. New York: Plenum Press.
3. Samet, M.G., Weltman, G. and Davis, K.B. 1976. Application of Adaptive Models to Information Selection in C3 Systems. Woodland Hills, CA: Perceptronics Inc., Report No. PTR-1033-76-12, December.
4. Siegel, A.I., Wolf, J.J. and Leahy, W.R. 1977. A Digital Simulation Model of Message Handling in the Tactical Operations System I. Its Sensitivity and User's Manual. Alexandria, VA: Army Research Institute, Report No. TR-77-A23, October.
5. Wohl, J.G. 1979. Battle Management Decisions in Air Force Tactical Command and Control. Bedford, MA: The MITRE Corporation, Report No. M79-233, December.

6. Samet, M.G. and Geiselman, R.E. 1980. Guideline Development for Summarization of Tactical Data. Alexandria, VA: Army Research Institute, Report No. TR-458, July.
7. Svanes, T. and Delaney, J.R. 1981. "SCAT: System Control Analysis and Training Simulator," in [Rasmussen and Rouse, 1981].
8. Callahan, C., Blum, R.W. and Witus, G. 1980. Guidelines for Managing the Flow of Information in an Automated Battlefield Command and Control System. Alexandria, VA: Army Research Institute, Report No. 1248, May.
9. Pfeuffer, F.J. 1980. "The Human Interface to the Switched Access Remote Test System," Bell System Technical Journal, Vol. 59, No. 4, pp. 529-556.
10. Dellner, W.J. 1981. "The User's Role in Automated Fault Detection and System Recovery," in [Rasmussen and Rouse, 1981].
11. Rasmussen, J. and Rouse, W.B. (Eds.), 1981. Human Detection and Diagnosis of System Failures. New York: Plenum Press.
12. Rouse, W.B. 1983. "Models of Human Problem Solving: Detection, Diagnosis, and Compensation for System Failures," Automatica (in press).
13. van Eekhout, J.M. and Rouse, W.B. 1981. "Human Errors in Detection, Diagnosis, and Compensation for Failures in the Engine Control Room of a Supertanker," IEEE Transactions on Systems, Man, and Cybernetics, Vol SMC-11, No. 12, December.
14. Rouse, W.B. and Morris, N.M. 1981a. "Studies of Human Problem Solving in a Process Control Task," Proceedings of the First Annual European Conference on Manual Control, Delft, The Netherlands, May.
15. Rouse, W.B. and Morris, N.M. 1981b. "Studies of Human Problem Solving in a Process Control Task," Proceedings of the 1981 International Conference on Cybernetics and Society. Atlanta, October.

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Bimonthly Report

For the Period October 1, 1982 - November 30, 1982

Contract MDA 903-82-C-0145  
(June 1, 1982 - July 1, 1985)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404/894-3996)

During this reporting period, the MABEL problem solving scenario was extended and refined to the point that an initial experiment should start soon, probably in January.

Now that the conceptual basis of MABEL has been formed, context-specific aspects of the problem solving scenario are becoming important. To gather this contextual information, initial contacts have been made in both the military (Ft. Gordon) and commercial (Southern Bell Telephone) communications domains. These contacts will be pursued over the next month to develop a good perspective of real-life control of large-scale communications networks. This information will be used to modify MABEL as necessary so as to reflect the types of decision that predominate in the military and/or commercial network control environments.

As of November 30, 1982, approximately \$20,000 had been spent.

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Bimonthly Report

For the Period December 1, 1982 - January 31, 1983

Contract MDA 903-82-C-0145  
(June 1, 1982 - July 1, 1985)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404/894-3996)

During this reporting period, information obtained from military communication networks (Ft. Gordon) and commercial networks (Southern Bell) was used to modify MABEL prior to experimentation. Fortunately, the modifications needed were relatively minor and pilot testing of the experimental conditions is now underway. Formal data collection should begin within one or two weeks. Independent variables under investigation include: nodes per display page, numbers of levels in hierarchy, and failure rate per node.

As of January 31, 1983, approximately \$30,000 had been spent.

CENTER FOR MAN-MACHINE SYSTEMS RESEARCH

SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING

ATLANTA, GEORGIA 30332

(404) 894-3996

April 4, 1983

Dr. Marshall Narva  
Army Research Institute  
5001 Eisenhower Avenue  
Alexandria, VA 22333

Dear Dr. Narva:

Enclosed please find four copies of our bimonthly report for the period of February 1, 1983 - March 31, 1983.

Sincerely,

William B. Rouse  
Professor and Director

WBR:j

CC: L. H. Bowman

F. Cochran

Encl.



HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Bimonthly Report

For the Period February 1, 1983 - March 31, 1983

Contract MDA 903-82-C-0145  
(June 1, 1982 - July 1, 1985)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404/894-3996)

During this reporting period, the first formal experiment with MABEL was completed. The independent variables were: number of nodes per display page, number of levels in the hierarchy, and failure rate per node. Twelve subjects each participated in nine sessions of approximately one-half hour in length. Subjects were crossed with numbers of nodes and levels, and nested with failure rate. Preliminary data analyses indicate that numbers of nodes and levels significantly affected performance while failure rate did not have a significant effect. Detailed analyses of this data will be pursued during the next reporting period.

As of March 31, 1983, approximately \$62,000 had been spent.

CENTER FOR MAN-MACHINE SYSTEMS RESEARCH

SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING

ATLANTA, GEORGIA 30332

(404) 894-3996

June 22, 1983

Dr. Marshall Narva  
Army Research Institute  
5001 Eisenhower Avenue  
Alexandria, VA 22333

Dear Dr. Narva:

Enclosed please find six copies of our annual interim report for the period of June 1, 1982 - May 31, 1983.

Sincerely,

William B. Rouse  
Professor and Director

WBR/gs

Enclosure

cc: F. Cochran  
OCA ✓ Research Services

# HUMAN PERFORMANCE IN MONITORING AND CONTROLLING HIERARCHICAL LARGE SCALE SYSTEMS

Richard L. Henneman and William B. Rouse

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, Georgia 30332 USA

## ABSTRACT

Human performance in monitoring and controlling activities in a hierarchical large scale network, such as a communications system, is considered. A scenario is described that is used in an experiment to examine three factors affecting humans functioning as network supervisor: cluster size (number of elements per display page), number of levels of pages in the hierarchy, and failure rate per element. Results indicate that increasing cluster size improves performance, increasing number of levels degrades performance, and failure rate affects only subjects' strategies.

## INTRODUCTION

Advances in computer technology are leading to the development of highly automated large-scale systems. Often these systems are structured as interconnected hierarchical networks consisting of a very large number of nodes and arcs. Examples exist in a variety of domains, including communications networks, transportation systems, and power distribution grids. Theoretical methods of performance optimization and control have been developed to cope with the complex environments these systems create [1,2]. Thus, many functions associated with the operation of these systems that were previously handled by a human can now be controlled by a computer. Unfortunately, there inevitably will be instances, such as during failure situations, that the computer will not be programmed to handle. At these times, theoretical approaches to control may break down, and a human is required to intervene and control the system in order to ameliorate the problem situation.

As has been argued previously [3], human abilities in dealing with these types of situation are not well understood. This lack of understanding is especially pronounced in the area of large scale systems. The dynamics of the system, its structure, and the human-system interface all combine to create tasks of possibly enormous complexity. Human performance in such tasks has not been investigated to any great extent.

Several issues surrounding human performance in these large systems are of particular importance. For instance, due to the large number of nodes in the system, only a limited amount of system information can be displayed to the human monitor at any one time. Thus, a trade-off exists between the number of items displayed to the human on each display page and the number of levels of pages in the hierarchy. Almost no guidance exists relative to what system configuration will result in the best human performance. Another set of issues surrounds the way humans access various portions of the system. While some work has been done in this area [4,5], it is not clear if performance will improve if humans scroll, window, or page through the system levels. Other important issues involve the human-system dialogue and performance aiding.

Underlying these issues is a need to investigate some of the basic factors that affect human performance in large scale systems. This paper proposes a means to assess these factors. In the next section an experimental scenario is described that incorporates important features of a typical large scale system. This scenario is used in an experiment to assess the effects of three large scale system parameters on human performance.

#### TASK DESCRIPTION

Before considering the scenario used in this experiment, it is worthwhile to discuss briefly a typical large scale system, namely, the nationwide telephone network of the Bell System. This example will establish a basis for the general discussion of the experimental task that follows.

The nationwide telephone system [6,7] is designed as a five-level hierarchical network composed of more than 170 million telephones and more than 22000 switching centers. The network consists of two basic elements: transmission and switching. The transmission elements are the actual communication paths that messages take through the system; the switching stations serve to economically interconnect calls.

A major feature of the system is its high degree of automation. Messages are sent through the system via direct or alternate paths that have been pre-determined. The system operates under normal conditions without any manual intervention. The switching stations, serving as repositories of network intelligence, automatically perform such tasks as : 1) determining source, sink, and path through the network; 2) testing lines for busy conditions before establishing a path;

and 3) continual checking of circuit conditions.

Nonetheless, human monitoring and maintaining of the system is still a necessity. During overload situations or in the case of major equipment failures, network performance can degrade rapidly. Human network controllers, thus, must intervene when the automatic solutions are excessively expensive or when a problem arises requiring human judgement. To deal with these situations, the human operator has at his disposal such tools as cancellation of alternate routing, reroutes, line load controls, and recorded announcements.

Using features of the Bell System as a general model\*, a computerized simulation of a generic large scale dynamic system was developed. This simulation is referred to as MABEL because of the obvious connotation, but also because it requires operators to Monitor, Access, Browse, and Evaluate Limits in the process of monitoring and controlling the system. MABEL is programmed in Pascal on a VAX 11/780 computer and operates in real time. It is structured as a large network that can range in size from hundreds to thousands of nodes. Customers travel through the system from a randomly selected source node to a random destination. Subjects monitor this system activity via a CRT display. When they detect a problem in the system (possibly due to a failure), subjects issue an appropriate command through a keyboard to correct and compensate for the abnormal situation. The overall system objectives are to maximize the number of customers served while minimizing the time they spend in the system.

The following sections discuss MABEL in more detail. Emphasis is placed on the structure of MABEL, the operator interface, and typical system operation.

### System structure

Several elements are basic to the structure of MABEL. A node represents the smallest structural unit in the network. Customers are processed at nodes with service times following an exponential distribution. Each customer is passed from node to node, following a path that will minimize its expected time in the system. If a node in a customer's path is currently busy, the customer joins a waiting line at that node until the node becomes idle.

As mentioned above, MABEL can consist of hundreds or thousands of nodes. It is impossible for the human operator to perceive and process information about all of the nodes at one time. On a more practical level, it is impossible to display

\*The simulation also parallels several aspects of military communications networks [8,9]. However, because automation is not as prevalent in the tactical military domain, the face validity of the simulation is higher for the current commercial network.

information about all of these nodes at one time. Thus, nodes are grouped into relatively small networks called clusters. Human operators are restricted to viewing only one cluster at a time on the MABEL display.

These clusters are grouped into hierarchical levels. A customer typically enters the system through a cluster in the lowest level. It proceeds through that cluster to a node that connects to a cluster in the next higher level. This process repeats until the customer reaches the top level of the system. At this point, the process reverses: the customer travels through a cluster and then "jumps" down to the next lower level. The process repeats until the customer reaches its destination.

Thus, as noted above, the system is analogous to a telephone communications system. Imagine a call being placed from Americus, GA to Mason City, IA. The message first travels from Americus to Macon to Atlanta via a network of telephone lines and switching stations. The message then travels from Atlanta to Chicago. It is then transferred to Davenport, IA, and finally proceeds to Mason City. Atlanta and Chicago are at the highest level of the hierarchical system; Macon and Davenport are at the second level, while Americus and Mason City are at the lowest level.

Customers initially arrive at a node in the lowest level in the system. These arrivals are scheduled according to a Poisson process. Routing through the system is completed automatically as determined by a shortest path algorithm. Thus, customers are routed through nodes that will minimize the time they spend in the system.

### Operator interface

Subjects obtain information about MABEL from a video display (Figure 1). The screen is divided into several sections. The upper right portion of the screen displays a cluster of nodes. The dim numbers to the left of each node identify the node, while the numbers inside each node represent the current queue size (the number of customers waiting to be served). This portion of the screen is updated approximately every two seconds. A different cluster of nodes is viewed by entering an appropriate command.

The lower right portion of the screen is an aid to the user to identify the current displayed cluster. Each letter (A,B,C) represents a level in the hierarchy. Each number (1,2,3,...) represents either a node or a cluster. Bright and dim characters are used to indicate the subject's current position in the hierarchy. A row of characters that is completely bright represents the cluster that is currently displayed on the screen. One bright character in a row of dim characters indicates the node above the currently displayed cluster. In Figure 1, therefore, the displayed cluster is in Level B. This cluster is beneath Node 7 of Level A.

The upper left portion of the screen is used to display the current time. As already noted, the time is updated approximately every two seconds. Since the system operates in real time, customers will keep arriving to the system whether any action is taken by the operator or not.

The middle left portion is used to display a variety of user-requested information about the system. This information is input at the prompt "Your action:", located at the lower left part of the screen. Ten different commands are available to the user. These can be grouped into four categories:

1. Commands that access cluster displays.
2. Commands that allow monitoring critical system variables.
3. Commands that provide diagnostic information by testing for failures.
4. Commands that control the system.

The ten different commands are summarized in Table 1.

#### Typical system operation

Under normal circumstances, MABEL will operate automatically without any interference from the human monitor. When a node failure occurs, however, the human must act to diagnose and repair it. Node failures can occur in two ways:

1. Total failure due to malfunctioning equipment: In this case a node is unable to service any customers waiting at it. All customers waiting at this node are lost, thereby reducing the queue size to zero. Additionally, the node refuses to accept any customers passed to it from another node. These customers are retained at their previous node. Since they are unable to proceed, the situation may lead to the following type of failure.
2. Failure due to exceeding the capacity of the node: Each node has a maximum number of customers that it can "store" at any one time -- that is, each node has a maximum queue size. If this queue size is exceeded, the node fails. Its behavior after this point is identical to equipment failures. The node is unable to accept customers, and thus, new customers are retained at their old node. Once a failure occurs, therefore, it is likely to lead to other failures. In the extreme case, if nothing is done to repair failed nodes, the entire system will fail.

It is, of course, also possible for this type of failure to be induced simply by trying to service too many customers: i.e., the system is trying to handle too much of the load. In this case, customers arrive at a node at a rate faster than the node can service them.



Subjects locate failures by monitoring critical system states and testing suspect nodes or clusters of nodes. If a failure is found, the subject dispatches a crew to repair the node. If the system becomes too crowded with customers, the subject can issue a command to reduce the number of customers admitted to the system.

## METHOD

### Subjects

Twelve subjects volunteered to participate in this experiment. All subjects were members of the research staff at the Center for Man-Machine Systems Research at Georgia Tech, with backgrounds in either engineering or psychology. Eight subjects were enrolled in graduate school, two were senior undergraduate students, and two were faculty.

### Training

Subjects were initially exposed to MABEL via a set of written instructions. These 19 pages of instructions contained a detailed explanation of the overall structure and normal operation of MABEL, a summary of the commands, and a section explaining the subject's role in operating MABEL during off-normal situations. Each section of the instructions concluded with a set of questions which were used to assess the subject's comprehension of the material in that section. The system and its operation was very complex to the novice; thus, these questions served to assure that subjects acquired a good understanding of MABEL.

Following the reading of these instructions, subjects were presented with a set of basic principles of operation and a summary of MABEL's structure. This material was organized and formatted for easy reference during MABEL operation. They were then given one final quiz to verify their understanding of the effects of failures on system performance.

The last part of the training session used a special version of MABEL that allowed subjects to stop the execution of the program at any time during the experimental run. This training program had the advantages of allowing subjects to become familiar with the commands and to become aware of the effects of failures on both display features and system performance, without being overwhelmed by the progressive effects of failures. If a situation became too complex, the subject could simply halt the dynamic system, solve the problem, and proceed. Subjects supervised two different training scenarios: a 16 node network with 2 levels, and a 9 node network with 3 levels.

## Experimental Design

Cluster size (i.e., number of nodes per display) and number of levels functioned as within-subject factors and failure rate served as a between-subjects factor. Cluster size varied between 4, 9, and 16; number of levels varied between 2 and 3. Failure rate was defined as the probability that a randomly selected node in the system would fail during each iteration of the MABEL program. One iteration occurred after each activity in the network (for example, the arrival of a new customer to the system). Failure rate was either low (probability of failure/iteration = .0005) or high (probability of failure/iteration = .001). The six subjects in each group saw six experimental conditions which consisted of all possible combinations of cluster size and number of levels. The order of presentation to subjects was balanced in order to average out any residual training effect.

## Performance Measures

A number of different performance measures were evaluated for each MABEL session. These measures can be classified as one of two types: product measures or process measures. Product measures assess the final result of a problem solving session, such as number of customers served. Process measures assess how that result was obtained by evaluating individual steps in a subject's strategic approach to supervising the system. In the following section, both product and process measures are discussed.

Product measures. These measures calculate the amount of time customers spend in the system and tabulate the number of customers served during an experimental run. Biases naturally exist in these measures simply due to inherent differences among the various MABEL systems. (For example, a 4 node network with 2 levels would be expected to process customers in less time than one with 16 nodes and 3 levels.) Thus, an effort was made to correct for this bias.

This correction was done by first developing an optimal solution for MABEL. This solution was derived by controlling MABEL with a computer program that had perfect knowledge of all system states. Thus, all failures were detected and repaired immediately. A set of optimal performance measures were collected from these sessions. Subjects' scores were then corrected by this optimal solution. Any deviation from optimality could, therefore, be accounted for by human limitations in dealing with the system. Measures developed using this approach were 1) average service time, and 2) number of customers served.

Process Measures. Three different sets of measures were classified as process measures: 1) errors, 2) failure diagnosis performance, and 3) strategy. The measures that composed each of these sets are discussed below.

Three measures of error were assessed. The first of these counted the number of times a subject saw a particular cluster of nodes that contained a failed node but did not repair the fault. The second error measure was actually a subset of the first: it counted the number of times a subject determined the presence of a failure via a test command but did not subsequently repair it. The third error measure counted the number of false alarms (repairing a node that had not failed).

A second set of measures assessed subjects' ability to locate failures. These measures were the average time to diagnose a failure and the percent of failures found in the system.

The third set of process measures looked at subjects' strategic approach to controlling MABEL. Measures included the average time between actions as well as the average amount of simulated time spent performing the four types of command activity (i.e., accessing, monitoring, diagnosing, and controlling).

## RESULTS

Analyses of variance were performed to determine the significance of the main fixed effects (cluster size, number of levels, and failure rate) on each of the twelve dependent measures discussed above. Subjects were treated as a random effect. The analyses were performed using the BMDP statistical library [10].

### Product Measures

Deviations from the optimal average time spent in the system were significantly affected by the cluster size and number of levels,  $F(2,20) = 8.06$ ,  $p < .005$  and  $F(1,10) = 20.30$ ,  $p < .005$ , respectively. As the number of levels increased from two to three, the corrected average time in the system increased from 0.96 seconds to 2.07 seconds. As the cluster size increased from 4 to 9 to 16 nodes, however, the corrected score decreased (from 2.26 seconds to 1.66 to 0.62). While a Newman-Keuls test indicated that there was no pair-wise difference between the means for four and nine node displays ( $p < .05$ ), the overall trend exhibited by this data is representative of many of the results found in this analysis. Subject performance degraded with increasing number of levels, while it improved with increasing display size.

Similar results were found from the ANOVA using the number of customers served (corrected for the optimal solution). In this case, however, the interaction of cluster size and number of levels had a significant effect on performance,  $F(2,20) = 4.70$ ,  $p < .03$ . Figure 2 indicates the nature of this interaction. Increasing the cluster size leads to improved performance at both two and three levels, with the difference in performance between levels most pronounced with four node networks. An analysis of

the simple main effect of number of levels with cluster size held constant indicated that performance means were different at 4 node and 9 node cluster sizes ( $F(1,10) = 8.87, p < .015$ ,  $F(1,10) = 6.54, p < .03$ , respectively), while there was no significant difference with the 16 node network. Apparently the number of levels plays a greater role in shaping performance when cluster size is smaller.

### Process Measures

Errors. Three ANOVAs were performed using the error measures. The first dependent variable was the number of times a subject viewed a cluster that contained a failure but did not take any repair action. This ANOVA indicated a significant main effect of cluster size,  $F(2,20) = 9.9, p < .001$ . As cluster size increased, the mean number of errors made by subjects decreased from 38.25 to 21.29 to 10.70. (A Newman-Keuls analysis indicated there was no reason to suspect a difference between the latter two means,  $p < .05$ .)

The next ANOVA used the number of tests made of a failed node with no subsequent repair action taken as the dependent variable. This analysis indicated a significant interaction between cluster size and number of levels,  $F(2,20) = 4.16, p < .04$  (Figure 3). An analysis of simple main effects indicated that only the nine node network produced a significant difference between performance at two and three levels,  $F(1,10) = 6.29, p < .035$ . Despite the large difference between mean estimates at the four node cluster size, this difference was not significant because of an extremely high level of variance associated with performance when monitoring the small system. Increased system size led not only to improved performance, but decreased performance variability as well.

The number of false alarms did not result in any significant main effects or interactions. Relatively few false alarms were made by subjects (the mean number was .53 per trial).

Failure diagnosis. Another set of performance measures evaluated subjects' performance in locating failures. The average time to diagnose a failure was significantly affected by the interaction of cluster size and number of levels,  $F(2,20) = 11.86, p < .0005$  (Figure 4). Increasing cluster size and number of levels led to longer times for diagnosis. Thus, despite the degradation of the system performance measures with decreasing cluster size, failure diagnosis time actually was less for smaller clusters. This may have been due to the fact that larger systems appear to be more tolerant of system failures (i.e., propagation of effects was slower).

Similar trends were indicated by the significant interaction between cluster size and number of levels in terms of the effect on percent failures found in the system,  $F(2,20) = 86.92, p < .0001$  (Figure 5). Perhaps surprisingly, even though subjects found relatively fewer failures in the larger systems, the

overall system performance as shown in Figures 2 through 4 was nevertheless better for the larger systems.

Strategy. An interesting question concerns whether the above performance differences were the result of inherent system differences or whether they were the result of subjects employing different control strategies. To answer this question, an analysis of several measures of strategy was performed. First, the average time between commands as the dependent measure resulted in a significant cluster size main effect,  $F(2,20) = 4.81$ ,  $p < .02$ . Subjects issued more commands with the smaller systems than the larger systems. The mean times in seconds for the 4, 9, and 16 node networks were 1.38, 1.71, and 1.77. (A Newman-Keuls analysis indicated no difference between the last two means,  $p < .05$ ).

The average time spent accessing parts of the system was significantly affected by the failure rate,  $F(1,10) = 5.08$ ,  $p < .05$ . Subjects in the low failure rate group spent about 18% of their time maneuvering through the system, while subjects in the high failure rate group spent about 27% of their time performing this activity.

Since subjects in the high failure rate group spent more time accessing the system, they apparently had less time to monitor the system. Failure rate significantly affected the amount of time spent monitoring:  $F(1,10) = 9.97$ ,  $p < .015$ . Means were 43.3% and 25.8% for low and high rate, respectively). The cluster size also affected the amount of monitoring,  $F(2,20) = 3.78$ ,  $p < .05$ . The means were 29.5%, 36.6%, and 37.7% for the 4, 9, and 16 node networks. A Newman-Keuls analysis showed no difference between the values for the 9 and 16 node networks.

The time spent performing diagnostic activities was affected by the interaction of cluster size and failure rate,  $F(2,20) = 4.49$ ,  $p < .025$  (Figure 6). Mean performance differed greatly between levels for the 4 node case, while as before it was similar for the 9 and 16 node case.

Finally, the amount of time spent controlling the system was significantly affected by all three variables: cluster size ( $F(2,20) = 13.55$ ,  $p < .0003$ ), number of levels ( $F(1,10) = 6.88$ ,  $p < .03$ ), and failure rate ( $F(1,10) = 7.39$ ,  $p < .025$ ). Increasing the failure rate increased the percent control time from 6.4% to 8.8%. Changing from two to three levels increased the percent control time from 6.4% to 8.8%, while increasing the cluster size decreased the percent time spent controlling from 11.0% to 7.3% to 4.9%.

Pearson product-moment correlations were calculated between the measures of strategy and the other performance measures. In order to perform the analyses, the measures were averaged across all six experimental sessions for each subject. Results indicated that performance was independent of subjects' strategic time allocation. Significant correlations ( $p < .01$ ) did exist,

however, between the percent time monitoring and the percent time accessing the system ( $r = -.955$ ) and between the percent time monitoring and the percent time diagnosing ( $r = -.837$ ).

These results suggest the prevalence of two basic strategies for supervisory control of MABEL. One strategy involved staying at higher levels and using monitor commands to assess the state of lower levels. The other strategy involved actually accessing the lower levels and performing tests to diagnose failures. Subjects with low failure rate conditions tended to choose the former strategy while subjects with high failure rate conditions tended toward the latter strategy. Apparently both strategies were effective in that performance was independent of either approach.

Figure 7 summarizes the difference in time allocation between the high and low failure rate groups. The results seem to agree with intuition, in that subjects with more failures to deal with would naturally be required to spend more of their time accessing parts of the system, diagnosing failures and controlling the system. The low failure rate group, on the other hand, would be enabled to monitor more often.

#### DISCUSSION AND CONCLUSIONS

The major determinants of subjects' performance in this task were the size of each displayed cluster and the number of levels in the system hierarchy. Often the interaction between these two variables was significant. Failure rate only affected the way subjects allocated their time among various activities. Perhaps the altered strategy of the high failure rate group enabled those subjects to maintain a level of performance consistent with that of the low failure rate group.

Increasing the number of levels tended to decrease the quality of performance. This effect was expected: the greater the percentage of nodes hidden from view, the greater the difficulty subjects experienced in supervising the system. For instance, the three level systems resulted in substantially longer times to diagnose failures. Since it took a longer length of time for the effects of lower-level failures to become obvious at the higher levels, the effects tended to be more serious than in the two level systems. This lengthened diagnosis time tended to degrade practically all other dimensions of performance.

A trend not predicted prior to the start of the experiment was that increasing cluster size would lead to improved performance. One would suspect that larger numbers of nodes per display should lead to increased task complexity. Thus, as the number of components that the human must deal with increases, performance should degrade. This is not the case in MABEL.

There exist two non-independent explanations for this phenomena. In order to normalize experimental conditions, each system was designed to process approximately the same number of

customers over the length of each experimental run. Moreover, each system was designed to be utilized to approximately the same degree. (For instance, each node in the top level of each system was busy approximately 50% of the time.) To achieve this uniformity, nodes in small systems had to have shorter mean service times than nodes in larger networks. Consequently, when a failure occurred, its effects became apparent quite rapidly. (Recall the shorter diagnosis times for small systems.) Unlike the large systems, therefore, one failure in the small system could quickly produce devastating effects to overall system performance due to the relative speed with which failures could propagate.

The second reason is that each node in a small network represents a larger fraction of the total size of the system than does one node in a large network. In other words, when a node fails in a small network, a relatively large portion of the overall system fails. Due to this failure, there are fewer available channels through which customers can be rerouted. Thus, customers tend to be retained more frequently at nodes when they have fewer alternate paths through the system. This also leads to faster propagation of failures in the smaller systems.

Therefore, smaller clusters were more vulnerable to failures due to both experimental normalizations and inherently fewer resources to absorb the effects of failures. Yet, the interaction of cluster size and failure rate did not significantly affect overall performance. Apparently, the aforementioned strategy shift was sufficient to compensate for increased failure rate, but not sufficient to deal with the more resource-constrained networks.

Overall, this initial experiment with MABEL produced two results of particular interest. First, as evidenced by Figures 2 through 5, the effects of number of levels in a hierarchical display system can be very strong, producing up to five-fold degradations for a modest change from two to three levels. This phenomenon deserves careful further investigation.

The second result of note is that strategies can be adapted to compensate for some situations (i.e., increased failure rate), but apparently not for others (i.e., small clusters with constrained resources). Of course, since failure rate was a between-subjects variable while cluster size was a within-subjects variable, there is no evidence that any individual subject can adapt his strategy as conditions change. Further research is needed to determine whether individual adaptation is possible or if some form of aided adaptation is needed.

## ACKNOWLEDGEMENT

This research was supported by the Army Research Institute for the Behavioral and Social Sciences under Contract No. MDA 903-82-C-0145.

## REFERENCES

1. M.D. Mesarovic, D. Macko, and Y. Takahara, Theory of Hierarchical, Multilevel Systems. New York: Academic Press, 1970.
2. Proceedings IEEE 1982 International Large Scale Systems Symposium. Norfolk, VA, October 1982.
3. J. Rasmussen and W.B. Rouse (Eds.), Human Detection and Diagnosis of System Failures, New York: Plenum Press, 1981.
4. K.D. Duncan, "Display ergonomics and fault location", Proceedings 2nd European Annual Manual. Bonn, 1982.
5. K.F. Bury, J.M. Boyle, and A.S. Neal, "Windowing versus scrolling on a visual display terminal", Human Factors, v.24, n.4, pp.385-394, August 1982.
6. Connecting a Nation. Administrative Training Center, Morristown, N.J., American Telephone and Telegraph Company.
7. Notes on the Network, Section 11: Network Management. Administrative Training Center, Morristown, N.J., American Telephone and Telegraph Company.
8. Combat Communications Within the Division, Field Manual 11-50, Department of the Army, March 1977.
9. Communications-Electronics Management System, Field Manual 24-22, Department of the Army, June 1977.
10. BMDP Statistical Software 1981, University of California Press, Los Angeles, CA: 1981.



Time = 417.21		
<b>System Statistics</b>		
	current time	% change
	358.53	337.69
Customers	96	-15%
Avg. Time	12.36	-3%
% Lost	2%	-50%
% Denied	1%	36%
Your action:		

1	3	5	3	9	3	13	
2		6	4	10	3	14	
3	2	7	2	11		15	1
4	1	8	8	12		16	

A:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
B:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
C:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Figure 1: MABEL Display

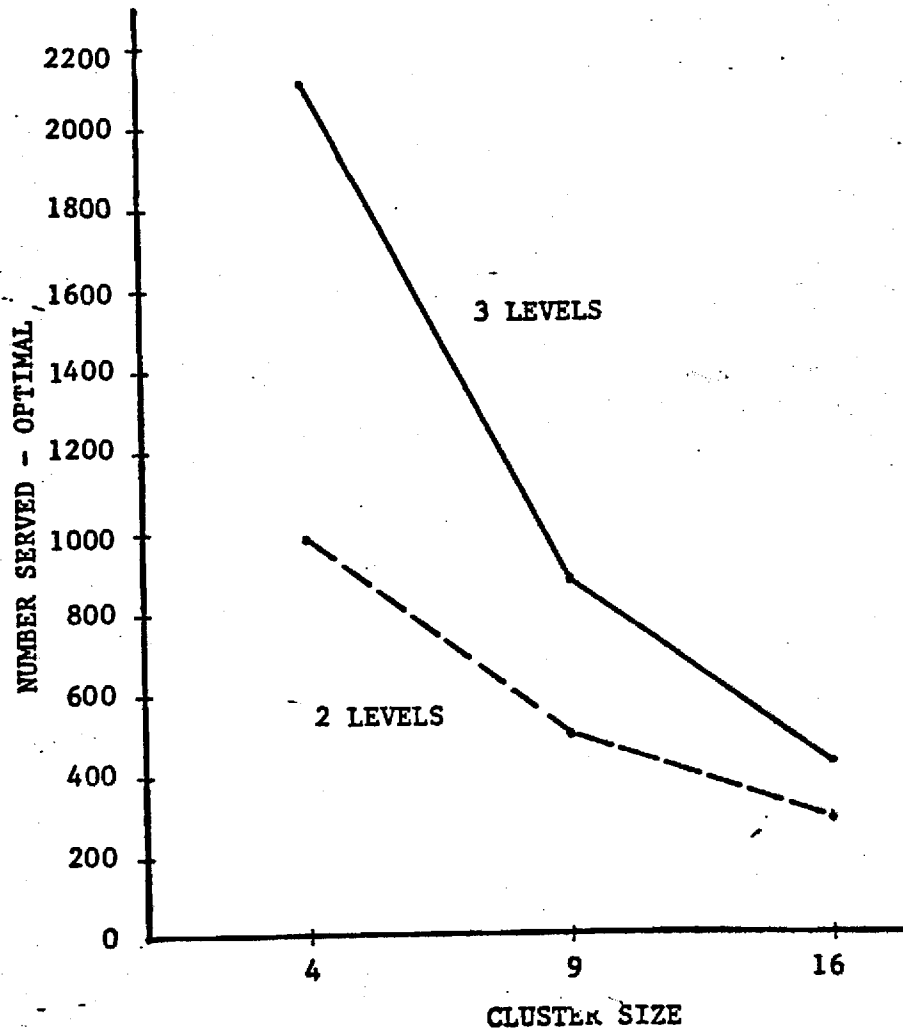


Figure 2: Number Served - Optimal vs. Cluster Size (at 2 and 3 levels)

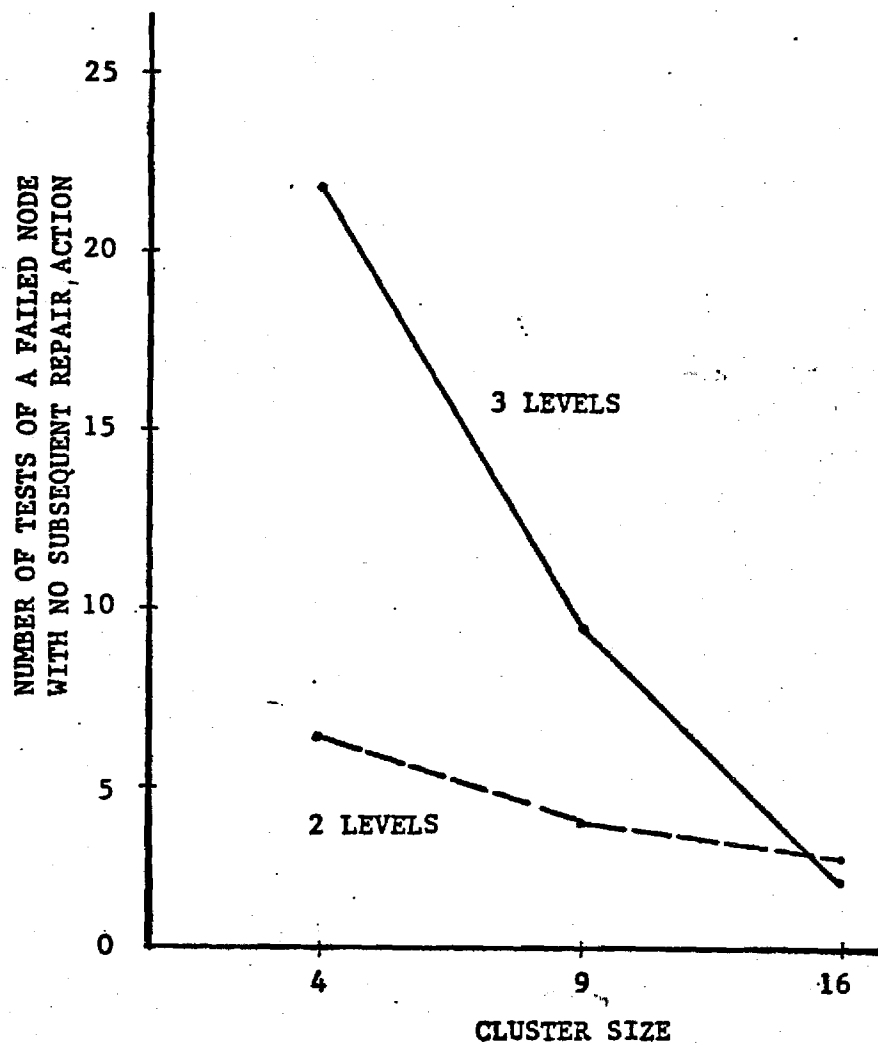


Figure 3: Number of Tests of a Failed Node with No Subsequent Repair Action vs. Cluster Size (at 2 and 3 Levels)

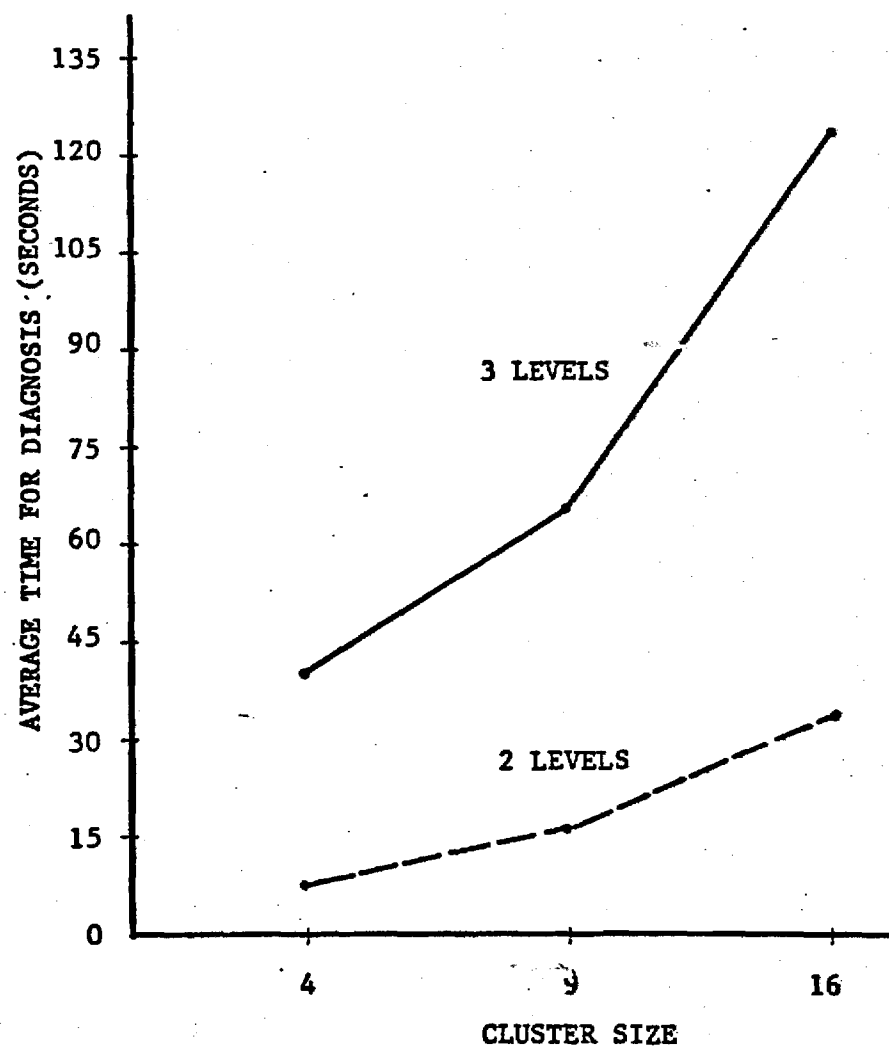


Figure 4: Average Time for Diagnosis vs. Cluster Size  
(at 2 and 3 Levels)

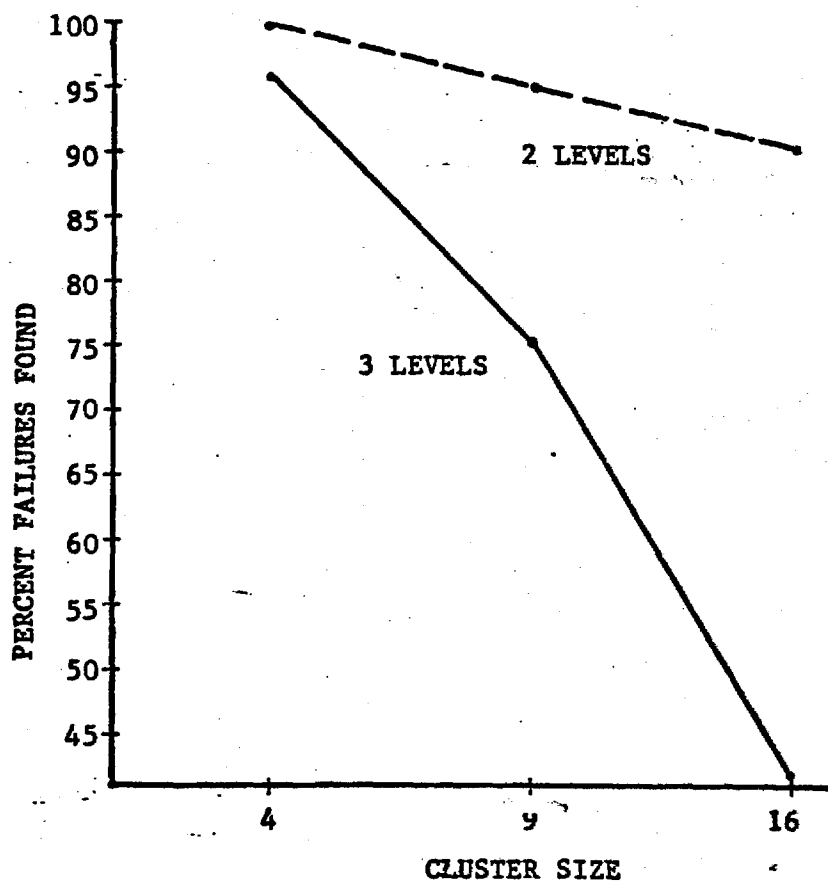


Figure 5: Percent of Failures Found vs. Cluster Size (at 2 and 3 Levels)

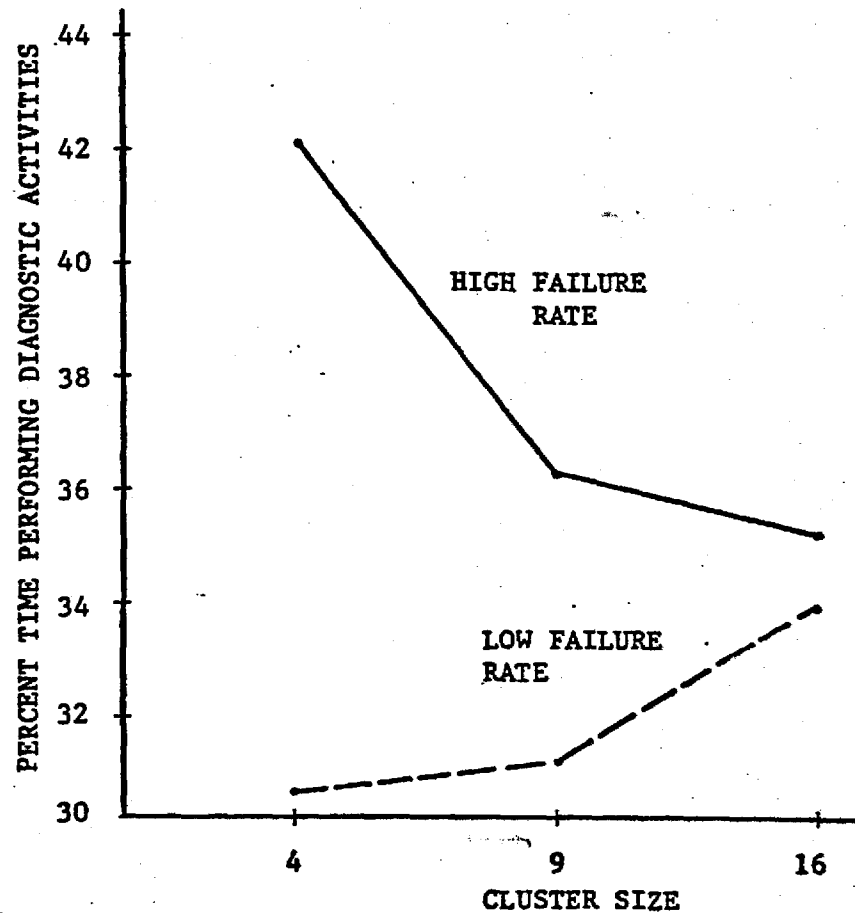


Figure 6: Percent Time Performing Diagnostic Activities vs. Cluster Size (at High and Low Failure Rates)

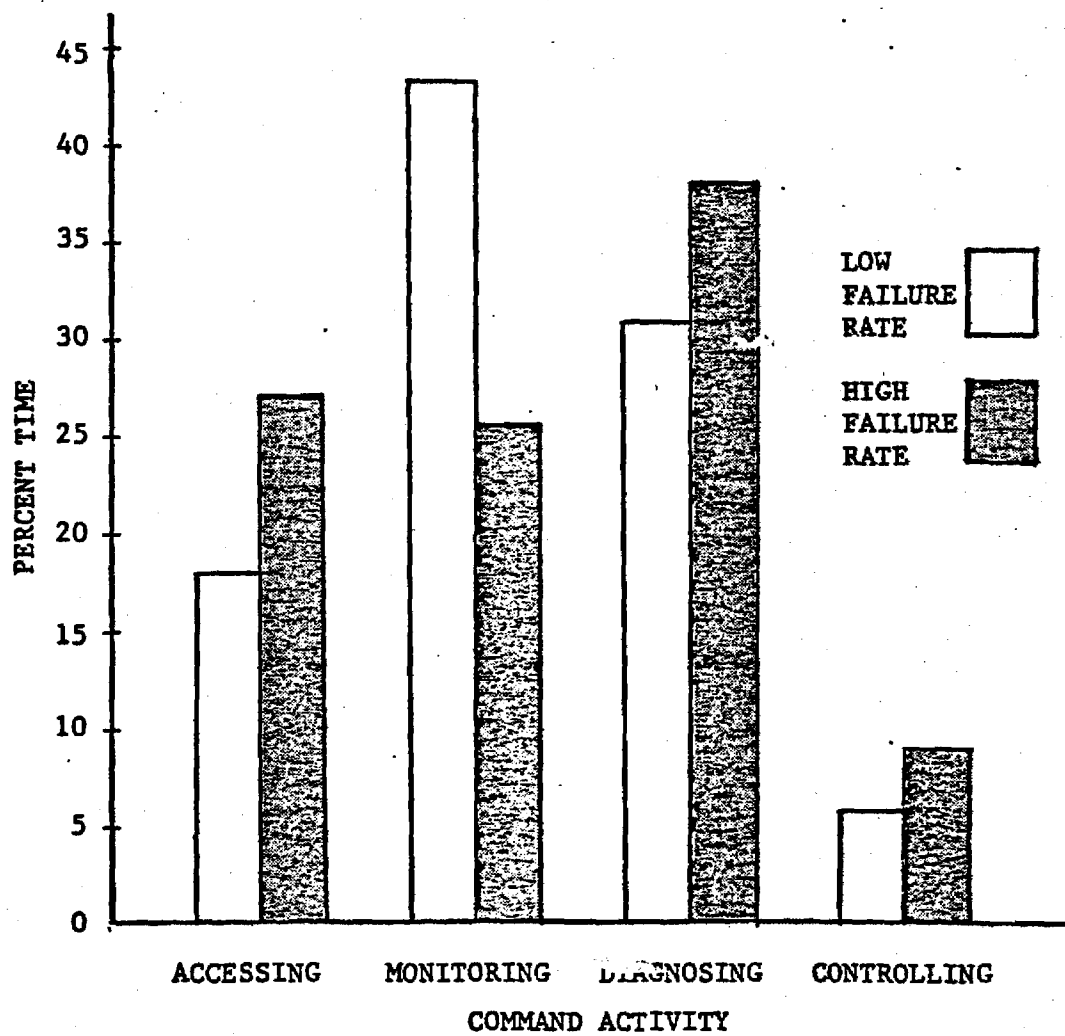


Figure 7: Percent Time Performing Command Activities

#### ACCESS COMMANDS

d	down one level
u	up one level

#### MONITOR COMMANDS

m	monitor next level
s	system summary statistics
c	cluster summary statistics

#### DIAGNOSTIC COMMANDS

t	test cluster for failures
n	node information

#### CONTROL COMMANDS

r	replace node
l	reduce load

Table 1: MABEL Command Set



CENTER FOR MAN-MACHINE SYSTEMS RESEARCH

SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING

ATLANTA, GEORGIA 30332

(404) 894-3998

August 24, 1983

Dr. Marshall Narva  
Army Research Institute  
5001 Eisenhower Avenue  
Alexandria, VA 22333

Dear Dr. Narva:

Enclosed please find four copies of our bimonthly report  
for the period of June 1, 1983 - July 31, 1983.

Sincerely,

*William B. Rouse*

William B. Rouse  
Professor and Director

WBR:de

cc: L.H. Bowman

F. Cochran

Encl.

*To: Res. Serv.  
Deliverable #7  
E-24-655*

*For  
8/26/83  
J*

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Bimonthly Report

For the Period June 1, 1983 - July 31, 1983

Contract MDA 903-82-C-0145  
(June 1, 1982 - July 1, 1985)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404/894-3996)

During this reporting period, the results of the first formal experiment with MABEL (reported in our Annual Interim Report) were used as a basis for beginning three new directions.

The first involved a modification of MABEL to include a geographical context involving regions, cities, and local exchanges in a telephone network. A small experiment (N=4) is currently being performed by Dick Henneman, to assess the effects of this context.

Dick Henneman is also beginning to test various measures of task complexity relative to the problem solving and control aspects of MABEL. Initial analyses of correlations between the uncertainty associated with each display page, which is strategy-dependent, and task performance look promising. A literature search on large-scale systems and complexity is also underway.

The third new direction initiated during this reporting period was the development of a rule-based model of human problem solving with MABEL. Eduardo Viteri is pursuing this work as his M.S. thesis. In order to expedite progress on this effort, which should have been initiated earlier but lacked staffing, Annette Knaeuper who is a new member of the Center's professional research staff will be working with Eduardo.

As of July 31, 1983, approximately \$90,000 had been spent.

# CENTER FOR MAN-MACHINE SYSTEMS RESEARCH

SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING

ATLANTA, GEORGIA 30332

(404) 894-3996

February 8, 1984

Dr. Marshall Narva  
Army Research Institute  
5001 Eisenhower Avenue  
Alexandria, VA 22333

Dear Dr. Narva:

Enclosed please find four copies of our bimonthly report for the period of October 1, 1983-November 30, 1983.

Sincerely,

William B. Rouse  
Professor and Director

WBR:paj

cc: L. H. Bowman ✓  
F. Cochran

Enclosures

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Bimonthly Report

For the Period October 1, 1983 - November 30, 1983

Contract MDA 903-82-C-0145  
(June 1, 1982 - July 1, 1985)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404/894-3996)

During the October-November reporting period, Dick Henneman's study of complexity progressed. A draft review of the complexity literature was completed and will be included in his Ph.D. thesis. Based on this review, as well as the characteristics of MABEL, work continued on developing measures of complexity for large-scale, dynamic networks. In October, Dick presented a paper on this work at the 27th Annual Meeting of the Human Factors Society in Norfolk.

Eduardo Viteri and Annette Knaeuper continued their work on developing a rule-based model of human problem solving with MABEL. The model structure is based on a hypothesis that emerged from an earlier ARI grant on diagnostic behavior. The model has been flow charted and programming has begun.

As of November 30, 1983, approximately \$118,000 had been spent.

# CENTER FOR MAN-MACHINE SYSTEMS RESEARCH

SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING

ATLANTA, GEORGIA 30332

(404) 894-3996

February 8, 1984

Dr. Marshall Narva  
Army Research Institute  
5001 Eisenhower Avenue  
Alexandria, VA 22333

Dear Dr. Narva:

Enclosed please find four copies of our bimonthly report for the period of December 1, 1983-January 31, 1984.

Sincerely,

William B. Rouse  
Professor and Director

WBR:paj

cc: L. H. Bowman ✓  
F. Cochran

Enclosures

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Bimonthly Report

For the Period December 1, 1983 - January 31, 1984

Contract MDA 903-82-C-0145  
(June 1, 1982 - July 1, 1985)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404/894-3996)



During the December-January reporting period, Dick Henneman developed a candidate complexity measure for MABEL that includes two sets of attributes that reflect: 1) system characteristics (e.g., redundancy), and 2) behavioral characteristics (e.g., strategy). Comparisons of this measure with data from the first experiment with MABEL produced fairly good results.

Also during this reporting period, a new version of MABEL was developed. CAIN (Contextually Augmented Integrated Networks) is structurally identical to MABEL, but is heavily laden with the context of a telephone network scenario including geographical labeling of clusters, demands for services varying with time zones, and outages that propagate locally. The purpose of adding the context is to make the task more meaningful for subjects and therefore encourage richer strategies.

Eduardo Viteri and Annette Knaeuper continued programming the model of human problem solving with MABEL. Currently, the issue being addressed is communicating the state of the network from MABEL to the model, and commands from the model to MABEL.

As of January 31, 1984, approximately \$138,000 has been spent.

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Annual Interim Report

For the Period June 1, 1983 - May 31, 1984

Contract MDA 903-82-C-0145  
(June 1, 1982 - July 1, 1985)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404/894-3996)

## INTRODUCTION

Current trends in computer and communications technology are leading to the development of many highly integrated systems in the domains of communications, transportation, manufacturing, etc. Most of these systems can be represented as large networks of nodes and arcs where nodes denote people, destinations, or machines and arcs denote communication lines, transportation routes, or a variety of activities. Because these systems are highly integrated, it is not unusual for there to be hundreds or thousands of nodes and arcs. Networks of this size and level of connectivity are very complex systems.

Complexity is further increased by the dynamic nature of these networks. The states of the nodes and arcs (i.e., levels flows, etc.) usually evolve in time and are not amenable to instantaneous control. Further, the demands placed upon the networks are often time-varying, with occurrences of peak demands not always being predictable.

This program of research is concerned with the problem solving behavior of the human whose role is network controller or operator. The job of the network controller is to manage the assets of the network (i.e., nodes and arcs) so as to maximize network efficiency. Further, during peak demand periods, the controller may have to implement control procedures such as load

shedding and priority scheduling to assure that overloads do not degrade network performance.

For many aspects of this job, the network controller has computer aids or, in fact, may simply have to monitor an automated system which performs many of the above functions. However, system failures or unusual environmental demands can require that the human intervene and manually control the network. The human's abilities to solve these types of problem are not well understood. In fact, human problem solving in complex dynamic environments is an area where few research results are available. This area is the topic of the research program whose progress is reported here.

## PROGRESS

This section briefly summarizes progress during the first two years of this three-year program of research. Considerable more detail can be found in the papers included in the Appendix.

Most of the first year was devoted to developing an experimental scenario and evaluating the impact of its parameters on human problem solving performance [Henneman and Rouse, 1984a]. Communications networks were chosen as the experimental context. After reviewing a variety of documentation on human control tasks in both commercial and military communications networks, an experimental scenario called MABEL was designed and programmed. MABEL requires subjects to monitor a large-scale automated communications network via a hierarchical multi-page CRT display. Much as discussed in the Introduction, subjects have to manage network assets and, in the event of a failure, intervene to diagnose the failure, compensate for its impact, and restore normal operation.

For the first formal experiment with MABEL, the effects of three independent variables were studied: 1) number of nodes per display, 2) number of levels in the display hierarchy, and 3) failure rate per node. Twelve subjects each participated in six experimental sessions. Overall, this initial experiment with MABEL produced two results of particular interest. First, the

effects of number of levels in the hierarchy were often very strong, producing up to a five-fold degradation of performance for a modest change from two to three levels. The second result of note is that rather different strategies seemed best for different combinations of independent variables. This leads to the question of whether humans can be trained to adapt appropriately or if some form of aided adaptation is needed.

The second year of this research involved two efforts. One effort concerned the development of a rule-based model of human problem solving in the MABEL environment. Eduardo Viteri pursued this modeling work in conjunction with his M.S. thesis. A summary of the model and a comparison of its behavior with that of subjects is included in the Appendix.

One general impression that emerged from the experiment and modeling efforts noted above was that MABEL lacked the contextual richness necessary to provide the type of problem solving environment required for this research. Perhaps the best indication of this is the simplicity of Viteri's model even though it compares fairly well with subjects' behavior.

This observation led to a decision to enhance substantially the contextual aspects of MABEL. A new experimental scenario emerged, CAIN for Contextually-Augmented Integrated Network. CAIN is structurally equivalent to MABEL, but has been augmented

to include many of the contextual elements of the telephone system in the United States (e.g., geography and effects of time of day, weather, and even construction projects). CAIN is summarized in a paper in the Appendix.

The first formal experiment with CAIN considered the effects of number of levels in the display hierarchy and level of network redundancy [Henneman and Rouse, 1984b]. Eight subjects each participated in thirteen experimental sessions. The purpose of having this large number of sessions was to study the effects of practice on strategies employed and resulting performance.

While the data analysis is only partially completed, several interesting results have emerged. First, similar to the experiment with MABEL, increasing the number of levels in the display hierarchy led to a degradation of failure diagnosis performance. Second, a fine-grained analysis of command usage by subjects indicated that rather disparate strategies can produce equivalent overall performance. Finally, a measure of problem complexity that includes both structural and strategic elements was found to correlate highly with diagnostic performance. Alternative interpretations of these results are currently being explored.

## REFERENCES

Henneman, R.L. and Rouse, W.B. "Human problem solving in large-scale networks," Proceedings of the 1982 International Large-Scale Systems Symposium, Virginia Beach, October 1982, pp. 293-297.

Henneman, R.L. and Rouse, W.B. "Human performance in monitoring and controlling hierarchical large-scale systems," Proceedings of the 27th Annual Meeting of the Human Factors Society, Norfolk, Virginia, October 1983, pp. 685-689.

Henneman, R.L. and Rouse, W.B. "Human performance in monitoring and controlling hierarchical large-scale systems," IEEE Transactions on Systems, Man, and Cybernetics, Vol SMC-14, No. 2, pp. 184-191, March/April 1984.

Henneman, R.L. and Rouse, W.B. "Assessing the complexity of a large-scale system: measures of system structure and human strategy," Proceedings of 1984 IEEE International Conference on Systems, Man, and Cybernetics, Halifax, Nova Scotia, October 1984.

Henneman, R.L. Human Problem Solving in Complex, Hierarchical Large-Scale Systems, Ph.D. Thesis in progress, Georgia Institute of Technology, 1984.



APPENDIX

ASSESSING THE COMPLEXITY OF A LARGE SCALE SYSTEM:  
MEASURES OF SYSTEM STRUCTURE AND HUMAN STRATEGY

Richard L. Henneman and William B. Rouse

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332

Abstract

The role of humans in monitoring and controlling activities in a hierarchical large-scale system, such as a communication network, is considered. An experiment was conducted in which subjects were required to monitor and control a large simulated network. The major control activity consisted of detecting and repairing system failures and relieving network overcrowding. Factors suspected of contributing to task complexity (e.g., number of hierarchical levels and degree of network connectivity) were varied. Data were analyzed from three perspectives, namely an analysis of variance of the effects of the independent variables, a fine-grained analysis of subject strategies, and an on-going development of measures of task complexity. It is proposed that the complexity of an operator's task is related to both system defined elements and the human's understanding of the system as reflected by his strategy. Time series analysis was used to identify transfer functions between the two types of complexity (structural and strategic) and the average time to failure diagnosis. Preliminary results suggest that the distinction between structural and strategic complexity is appropriate.

## I. Introduction

Recent technological advances have enabled the development of highly automated large scale systems. These systems frequently are represented as networks that consist of a number of nodes connected by arcs. In addition, these systems often are arranged hierarchically, with the extent of control increasing with successive hierarchic levels. The network size, degree of integration, and hierarchic structure all combine to create systems of potentially enormous complexity. The complexity is compounded by the fact that these systems are dynamic -- the states of the arcs and nodes evolve over time and are not amenable to instantaneous control. Examples of this type of system exist in many domains. Commercial and military communication networks, and transportation, manufacturing, and power systems can all be represented as large hierarchic dynamic networks.

Of major concern is the ability of people to monitor and control these networks. Despite these systems' high level of automation, human intervention and control is necessary when unexpected events occur, such as system failures or network overcrowding. The research addressed within this paper considers these human abilities.

This paper is divided into seven sections. In the next section, an experimental scenario is described which was used to investigate the relationship between large scale system complexity and human performance. The scenario extends the context-free, large-scale simulation reported in Henneman and Rouse [3] by introducing a high level of contextual detail. Section III describes an experiment that was conducted using this simulation, while Sections IV, V, and VI analyze and discuss results from this experiment. Section IV includes the results of an analysis of variance of the independent experimental variables. Section V examines the experimental data on a more detailed level by making inferences relative to subject strategy from the analysis of individual commands. Section VI discusses the complexity of large scale systems and its implications for human performance. Finally, Section VII summarizes the results of this research.

## II. Description of Experimental Scenario

A previous experiment [3] considered human performance in the monitoring and control of a generic large scale system. Subjects monitored activity within a relatively context-free simulated environment referred to as MABEL (due both to the obvious connotation with the nationwide telephone network and the fact that it required subjects to Monitor, Access, Browse, and Evaluate Limits in the process of controlling the system). The network is structured as a hierarchical network composed of nodes, clusters, and levels. MABEL is programmed in Pascal and operates on a VAX 11/780 computer. The simulation is relatively context-free in that it can represent any of several large-scale domains (e.g., manufacturing system, transportation network, communication network); subject training, however, emphasized the similarity to the nationwide phone network of the Bell System.

Of substantial theoretical interest is the extent to which human abilities in coping with the complexity created by this type of system is a function of the level of abstraction present in the human-system interface; i.e., is better performance in some situations facilitated through the addition of increased contextual information? While this question will not be addressed in this paper, the issue did provide partially the motivation for expanding the MABEL scenario to contain a higher level of contextual information. In addition, the contextually augmented scenario increases the experimental validity by increasing the simulation fidelity.

Henneman and Rouse [3] describe the MABEL system in detail. Below are summarized aspects of MABEL (i.e, the structure, interface, and typical system operation) that are common to both MABEL and the new scenario. The new scenario is then described in more detail.

### MABEL

The physical structure of MABEL is composed of a very large number of nodes. Customers are processed at nodes and passed on to other nodes, following a path that minimizes the time between their source and sink nodes. The system operates, therefore, as a very large queueing network.

Due to the size of the system, it is impractical and unnecessary to display all potentially relevant system information to the human operator. Thus, nodes are grouped into relatively small networks called clusters. Operators may view only one cluster of nodes at a time on the MABEL display. Clusters are grouped, in turn, into hierarchical levels.

During normal operation, human subjects monitor activity in the system via a CRT display. Since MABEL operates in real time, the critical system states (namely, the number of customers waiting at each node) change with time. This information is updated every three seconds on the display. In addition, a variety of other information pertinent to proper system functioning is displayed as requested by the user. Subjects communicate with the system by inputting commands via an alphanumeric keyboard.

Under normal circumstances, MABEL operates automatically without any direct control action by the human monitor. When a critical event occurs, however, such as a node failing or system overcrowding, the operator must first identify that the event has occurred and then issue corrective action. Control action in MABEL takes the form of either node repair or load reduction.

#### CAIN

MABEL was substantially altered to produce CAIN (Contextually Augmented Integrated Network). While the physical hierarchical structure of MABEL was preserved, the addition of contextual information necessitated changing certain features of the interface. In the MABEL scenario, for example, all nodes on any particular display page are identified by a number on the CRT display. Each displayed node in a cluster, therefore, looks physically the same as nodes in other clusters. The MABEL interface has a generic quality in that all system sections are physically similar; no contextual cues exist. On the other hand, system elements in CAIN are identified via specific geographic locations. In MABEL, for example, if a subject wanted to display a lower level cluster, he might input the command "d9", which would display the cluster beneath Node 9. In CAIN, on the other hand, the subject might type "dSanFranc", which would display the cities beneath San Francisco (e.g., Berkeley, San Jose). Thus, subjects could form associations or links between the system parts due to the existence of the contextual information.

Simply introducing geographic names as node labels is not enough, however, to facilitate a change or difference in subject task performance. A small experiment ( $n=3$ ) replicated the first MABEL experiment, with the exception that nodes were given geographic names. No significant difference existed in terms of performance between subjects using either of the two task scenarios. A possible explanation is that the additional context must be necessary to adequately perform the task, otherwise it is of no use to the operator. Thus, events must occur that are related to specific locations in the system.

Such context-dependent events were introduced to CAIN. Although equipment in nodes fails randomly, some equipment experiences a higher probability of failure. For example, a thunderstorm in Little Rock, AR may make equipment in that city susceptible to lightening damage. Similarly, vandals in Newark, NJ might be more apt to damage equipment than farmers living near Council Bluffs, IA. Therefore, equipment in certain cities exhibits a greater tendency to fail than in other cities. Subjects are informed of these locations via warning alarms that flash on the bottom of the display. Subjects can directly monitor activities within these trouble spots via a special "watch" command.

Besides recurring failures, another type of context-dependent event was introduced to CAIN. At different times, certain sections of the system may be more prone to experience heavy loading than other sections. For example, certain times of day are busier in one part of the country than in others. Similarly, a major political or sports event in one section of the country may increase the number of messages sent. As with the recurring failures, subjects are informed of the location of these fluctuating loads via messages at the bottom of the screen.

A final change that was implemented in CAIN involved the way in which subjects can access information about the various clusters. In MABEL, movement is constrained in that subjects can only display the cluster of nodes in a level immediately above or below the current display. Thus, it is not possible to "jump" laterally across the network. In CAIN, however, it is possible to move from any part of the system to any other part. Thus, if a subject recalls that the cluster associated with Bangor, ME was previously experiencing problems, it is relatively easy to call up that cluster display.

In summary, therefore, despite the structural isomorphism of the two simulations, CAIN represents a significant departure from the context-free scenario of MABEL. Through the addition of contextual detail and the addition of events that are dependent upon this contextual information, the level of fidelity of the simulation has been enhanced significantly.

### III. Experiment Two

#### Motivation

The main goal of Experiment Two was to investigate the nature of complexity in a large scale human-machine system. The general assumption was made that task complexity can only be measured relative to an individual's understanding of the system and his expertise in dealing with problem situations within that system. (More detail relative to complexity may be found in Section VI of this paper.) Thus, complexity is considered to be dynamic, varying across time and among subjects. Accordingly, subjects were required to perform the task (CAIN) over a relatively long period of time.

#### Subjects

Eight junior and senior engineering majors at Georgia Tech served as subjects in this experiment. Due to the nature of the task, potential subjects were screened via a typing test (minimum ability level was 25 words/minute). Subjects were paid a total of \$65: \$5.00 for each training session (3) and each experimental session (10).

#### Training

Subjects were trained via a combination of written instructions and hands-on experience with CAIN. Subjects initially were given two sets of written instructions on consecutive days explaining the system, the goals of their task, and methods for achieving these goals. Self-test questions were contained within the text to insure mastery of the material. The experimenter reviewed this material with each subject at the beginning of each training session. In addition, subjects were given one-page summaries detailing the structure of the system, operation of the system, and available commands.

Subjects completed each training session by controlling a two-level CAIN system. A third training session was spent controlling a three-level CAIN system. These sessions were performed using a version of CAIN that allowed subjects to start and stop the program execution. Thus, subjects could investigate normal and abnormal system functions without being overwhelmed by the progressive effects of failures. The experimenter was present during all training sessions to answer questions.

#### Experimental Design

Results from Experiment One suggested that the degree of interconnectivity between nodes in MABEL had a particularly strong effect on task performance. Another result from Experiment One showed the very strong effect of number of levels within the hierarchical system. Thus, two independent variables selected for Experiment Two were the number of levels in the system and the degree of redundancy. Redundancy varied between low (6 connections/node) and high (13 connections/node) and number of levels varied between two and three. Cluster size was kept constant at 16 in order to emphasize the non-varying features of the contextual display.

Of major interest in this experiment was the way in which complexity changes with increasing subject expertise. Thus, the order of presentation of experimental conditions to subjects was not randomized. All subjects saw the same experimental conditions in exactly the same order. A final independent variable, therefore, was the order of presentation of experimental conditions.

In summary, the experimental sessions (S1 - S10) were performed in the following order: S1,S2: 2 levels, high redundancy; S3,S4,S5: 3 levels, high redundancy; S6,S7: 2 levels, low redundancy; S8,S9,S10: 3 levels, low redundancy. Each experimental session was performed on consecutive days and lasted about 45 minutes.

#### IV. General Results from Experiment Two

Data files from Experiment Two were analyzed using the same performance measures as Experiment One [3]. Results from Experiment One led to the following conclusions. First, the major determinants of subject performance were the size of each displayed cluster and the number of levels in the system hierarchy. Often the interaction between these two variables was significant.



In general, increasing the number of levels tended to degrade the quality of performance, while increasing the cluster size tended to improve performance. Failure rate only affected the way subjects allocated their time among various activities.

Perhaps the most surprising result from the Experiment Two data is the relatively strong effect of number of levels. Although this finding in itself is not surprising, the direction of the effect is: both number of customers served and average time spent in the system were significantly affected by increasing the number of levels from two to three. Increasing the number of levels actually improved performance. While these results are disconcerting at first considering the strong effect in the opposite direction found in Experiment One, they may be explained in light of changes made in the experimental scenario.

Due to the addition of contextual information to CAIN, changes were necessarily made in the temporal sequencing of events existing in MABEL. Thus, the real time length of each experimental run increased as the number of system levels increased. This increase had the net effect of slowing down the dynamics in the three level systems. Thus, the three level system allowed subjects more time to make critical decisions. This increase in time apparently had the effect of improving subject performance in terms of global measures of performance.

It is important to note, however, that these findings only exist for the product measures of performance (e.g., number of customers served and average time in the system). When considering measures of the problem solving process the results are similar to those of Experiment One. For example, increasing the number of levels from 2 to 3 increased the average time to failure detection from 23.07s to 42.71s ( $F(1,7) = 76.65, p < .0001$ ). In addition, the percent of failures found decreased from .950 to .723 ( $F(1,7) = 529.54, p < .0001$ ). Increasing the number of levels, therefore, significantly decreases the ability of subjects to locate failures.

It can be argued that these process measures that are related to failure diagnosis ability are better indices of subject performance than the product measures. Process measures are more direct metrics of subject ability in that they are assessing subject related characteristics rather than system-dependent

characteristics, such as number of customers served. From this perspective, the general results from Experiment Two are quite consistent with those from Experiment One.

The remaining independent variables, degree of redundancy and session, both affected performance as expected. In general, decreasing the degree of redundancy tended to degrade performance. If a node has fewer connections, less alternate routes existed for customers through the system, thereby leading to more capacity failures. In addition, subject performance tended to improve with increasing experience. For example, the average time to failure diagnosis decreased from 38.00s to 27.75s the last time a particular experimental condition was seen by subjects ( $F(1,7) = 57.05$ ,  $p < .0001$ ). This result is typical of many of the remaining performance measures; thus, they will not be reported here.

In summary, these analyses of variance yielded several useful results. Results confirmed hypotheses relative to the effect of the independent variables: subjects improved with experience, and as the number of connections per node decreased, subject performance degraded. A second main result confirmed results from Experiment One concerning the effect of number of levels on subject performance; those measures directly related to subjects' fault detection ability degraded with increasing number of levels.

#### V. Command Level Analysis

Beyond the general analysis completed above, it is of considerable interest to examine the strategies that people use in operating CAIN. In a dynamic task such as CAIN, an individual's strategy is not only dependent on what action is performed, but also when it is performed. Thus, an operator may perform a highly appropriate action in light of the current system state; if the action is not implemented soon enough, however, worse system performance may result.

The analysis that follows, therefore, considers both the nature and the timing of subjects' actions. First, the frequency of individual commands and pairs of commands will be tabulated. Second, the average real time between commands will be determined. These summaries will then be used both qualitatively and quantitatively in an analysis of strategy.

### Qualitative Strategy Analysis

Transition matrices were tabulated that counted the frequency of every possible command transition for each subject and for each experimental session. A cursory examination of these matrices yields several interesting results. First, wide variations exist between subjects simply in terms of the number of commands issued. For example, during Session 4 one subject issued 1655 commands while another issued only 874. Their performance in terms of average time to failure detection was approximately the same (67.64s vs. 67.28s). Since subjects could apparently achieve comparable results using very different numbers of commands, this observation suggests that several different strategies were appropriate for controlling CAIN.

Second, wide variations exist between types of system: in particular, subjects issued many more commands when operating a three-level system than when operating a two-level system. This observation is particularly relevant in light of the effects of number of levels found in Section IV. Third, wide variations exist between the frequency with which command types were used. Related to this result is the observation that less frequently used commands had longer latencies. This tradeoff concerning the degree of automaticity of command use will be considered more quantitatively in the next section. Finally, wide variations exist between usage of sequences of commands.

The way in which these strategies differ may be explained by considering two apparent "dimensions" of strategy. The first dimension involves the manner in which information seeking is pursued by the operator: does the subject actively seek information or passively observe the system state? In general, a subject's position along this continuum should be related to his overall frequency of command use. A subject who issues many commands can be considered a more active information seeker than one who issues few commands.

The second dimension involves the way in which subjects make decisions relative to proceeding down a "path" in the network. A subject may explore, for example, a lower system level because of an observed critical system variable noted on the information display (i.e., a large queue size). On the other hand, a subject may explore a lower system level because of information gathered from a command that displays data about the activity in the next lower level. The former approach relies on inferring the lower system state on the

basis of upper level clues, while the latter strategy makes inferences based on directly observing the lower system state.\*

The degree to which a subject uses one approach or the other may be estimated by examining the frequency of certain command sequences. In CAIN, a common command sequence consists of displaying a cluster in the level beneath the currently displayed level (a "d" command), and testing the new cluster for failures (a "t" command). If this "d-t" command sequence is preceded by the command that allows direct monitoring of the lower system state ("m" - monitor command), it suggests that the subject is examining the next level due to an observation made from the monitor command. Thus, the extent to which a subject uses a monitoring approach rather than an inferential approach may be estimated by calculating the percent of times that a subject preceded a "d-t" sequence with an "m".

#### Quantitative Analysis

Besides the frequency of command use, subject strategy is also related to the time needed to issue a command. These times vary as a function of command type (e.g., a frequently used command typically takes less time to issue than a less used command) and also with subjects.

With these observations in mind, graphs which plotted command frequency vs. inter-command time (or command latency) were developed. These plots were considered for each session and for each subject. The general result was that the plotted points tend to fall along a negative diagonal (as expected) with one significant exception: the "down" command, typically associated with a high frequency of use, also tends to have relatively long latencies. This command appears to be related to an aspect of strategy different from the other commands. It can be argued that the critical task in CAIN involves deciding when to examine a lower level in the system. Thus, it is reasonable that the inter-command time associated with the down command is longer than would be predicted otherwise. The analysis proceeded by performing a simple linear regression to estimate the slope of the line relating command frequency to command latency (omitting the down command).

\*Of course, subjects using this direct assessment approach could also have used the inferential strategy as that information was always available on the display.

These slopes were then used with "d"-command frequency in a regression analysis to predict some of the global measures of subject performance, namely, number of customers served (corrected by the optimal), average service time (corrected by the optimal), and average time to failure detection. Table 1 summarizes the regression equations, which are based on data from all sessions and all subjects.

Regression analyses were also performed using subsets of the original data to determine the degree to which the equations are affected by the experimental variables (i.e., number of levels and degree of redundancy). Redundancy had no effect on the regression equations in terms of both significance of the model and the amount of variance explained. In addition, all model coefficients were of the same order of magnitude.

On the other hand, using data from only the two and three level systems lowered the percent variance explained and the overall significance of the model. Nevertheless, no significant difference existed in terms of the regression coefficients for either the number served or average service time equations. In terms of the equations for average time until failure detection, however, a significant difference did exist between the two sets of coefficients. While the equation for the three level system was not significant, the two level system resulted in an equation with  $F(3,74) = 21.10$ ,  $p < .0001$ , and  $R = .770$ , a large improvement over the combined model. The reason for the improvement is probably that failure detection times for the three level systems are much more variable than those for the two level systems; thus, these times are more difficult to predict.

In each equation, the coefficients may be interpreted as follows: as the slope becomes more negative, performance improves. In addition, as the frequency of down commands decreases, performance improves. These results suggest that best performance was achieved by subjects who issued relatively few "down" commands, but tended to issue other commands more frequently and/or quickly (thereby increasing the slope).

The extent to which subjects achieve varying levels of performance relative to this apparent tradeoff between command frequency and latency was analyzed by estimating for each subject and session the x- and y-intercepts of the individual regression equations. By examining the standard deviation of

the mean of these values across all sessions (i.e., the intercepts), it should be possible to estimate the extent to which frequency or latency contributes to the slope.

The analysis showed that the y-intercept (corresponding to command frequency) was 132.69 with a standard deviation of 61.08, resulting in a coefficient of variation of .460. The x-intercept (corresponding to command latency) was 6.986, with a standard deviation of 1.866, resulting in a coefficient of variation of .267. The results indicate that there is considerably less variability in the command latency measure (.267) than command frequency measure (.460). Thus, these results suggest that subjects are enabled to gain more information about the system by issuing more commands other than down commands. This gain in information is reflected by an improvement in subject performance.

#### VI. Assessing the Complexity of a Large Scale System

As mentioned in Section III, the main purpose of this experiment was to examine the nature of complexity in a large scale system and its impact on human performance. While this analysis has not been completed, this section serves as an introduction to the underlying premise of the approach, as well as a description of the statistical methodology used to analyze the data. Time series analysis was used to develop transfer functions relating the two complexity measures to the average time until failure repair.

##### Large Scale System Complexity

A review of the literature [4] has suggested that the complexity of a large scale system may be described in terms of 1) the physical structure of the system and 2) the operator's understanding of the system as reflected by his strategy. From this perspective, a system that is complex or difficult to control for one supervisor may be relatively easy to control for another supervisor. Similarly, the complexity of any particular system may vary with time for any particular operator. Some systems, however, may be complex regardless of any particular control strategy due to their inherent structural complexity.

The measure of structural complexity is estimated by calculating the total number of display pages the subject must view in order to repair all failures in the system. If the subject performs perfectly, this measure represents the minimum number of pages necessary to discover all failures. Thus, the structural complexity measure represents optimal performance given the constraints of the structure or arrangement of the system components. This measure is only affected by subject performance in that any given subject may have more or fewer failures in the system depending upon their fault finding ability.

The strategic complexity measure, on the other hand, explicitly considers the subject's performance. When a subject is deciding which "path" through the system is most likely to lead to finding a failure, he makes a tradeoff between the time since his last observation of that part of the system and his expectations of finding a failure in that part of the system. High uncertainty about a part of the system may be acceptable, for example, if a relatively low probability exists of finding a failure in that section. The converse also is true. The measure of strategic complexity multiplies these two metrics (state uncertainty and probability of finding a failure given the system state) and sums them across the entire system.

The literature review [4] has further suggested that an appropriate dependent measure of complexity is the time until failure repair. Thus, the two independent measures of complexity (structural and strategic) were combined in one equation to predict the time until failure repair. More specifically, the dependent complexity measure is the average time until the subject issued a repair command for a failed node. The average includes only the five previous repairs.

#### Development of Transfer Functions

Since the dynamics of a large scale system are not instantaneous, the effects of complexity at any given time may not manifest themselves for some time lag. Thus, time series analysis was identified as a suitable means of developing transfer functions that relate the two complexity measures at various time lags to the current value of the average time until repair.

The basic methodology is that espoused by Box and Jenkins [1]. Carter [2] discusses the use of time series models in human factors research, while Montgomery and Weatherby [5] provide a good tutorial on the development of two independent variable transfer function models. Their approach is outlined in Henneman [4].

### Initial Results

Data for the time series analysis were generated by replaying subject data files. Following every 3 seconds of real time, both complexity measures and the average time until failure repair were calculated. These measures represented the two input and output time series. Results of this complexity analysis are not complete. In general, findings so far support the conclusion that the two complexity measures are related to the difficulty subjects have in locating system failures. The transfer function models that use these measures explain between 85% and 95% of the variance within the data. A problem arises, however, in the interpretation of the transfer functions. In particular, it is not clear how to correctly interpret the meaning of the lags and coefficients in the equations. Current work is considering refinement of the measures and possible explanations of the equations.

### VII. Conclusions

In summary, the work reported in this paper represents an effort to understand human problem solving in the monitoring and control of large scale systems. The analysis of results has considered human performance in a simulated task at increasing levels of detail, ranging from global measures of problem solving performance and their relation to certain aspects of system design (e.g., number of levels and degree of redundancy), to a more fine-grained analysis of subject strategy. In addition, a means was proposed to relate the system structure and a subject's strategy to a measure of task complexity.

It should be emphasized that although bits and pieces of a global understanding of human problem solving in large scale systems are beginning to emerge from this research, it is premature to propose a unified conceptual approach to these results. In general, however, the results presented in this paper suggest that human performance in the monitoring and control of large scale systems can be severely limited by both the structure of the system



(i.e., number of levels) and the human's understanding of the system as reflected by his strategy.

#### References

1. Box, G.E.P., and Jenkins, G.M., Time Series Analysis: Forecasting and Control, San Francisco: Holden-Day, 1976.
2. Carter, R.C., "Time series models of human factors dynamics," Human Factors, v.26, n. 1, pp. 83-95, February 1984.
3. Henneman, R.L. and Rouse, W.B. "Human performance in monitoring and controlling hierarchical large-scale systems," IEEE Transactions on Systems Man, and Cybernetics, v. SMC-14, n. 2, pp. 184-191, March/April 1984.
4. Henneman, R.L. Human Problem Solving in Complex, Hierarchical Large-Scale Systems, Ph.D. Thesis in progress, Georgia Institute of Technology, 1984.
5. Montgomery, D.C. and Weatherby, G., "Modeling and forecasting time series using transfer function and intervention methods," AIIE Transactions, v. 12, n 4, pp. 289-306, December 1980.

MEASURE	INTERCEPT	SLOPE	D-COMMDS	F(2 77)	R
Average time to failure detection	= 23 27	+ 0 92	+ 0 20	25.58*	.632
Number of customers served (corr by optimal)	= 924 66	- 12 54	- 3 75	16.50*	548
Average time in system (corrected by optimal)	= 33 75	- 0 51	- 0 17	18 78*	573

\* p < .0001

Table 1: Summary of Regression Analyses

# A RULE-BASED MODEL OF HUMAN PROBLEM SOLVING FOR A DYNAMIC PLANT

Eduardo T. Viteri

August, 1984

## INTRODUCTION

A rule-based model has been developed to mimic a human operator of a complex communication network. The communication network is specified by MABEL (Monitor, Access, Browse, and Evaluate Limits). Very briefly, MABEL is a computerized simulation of a generic large scale dynamic environment very much like the AT&T telephone network. The objectives of the operator are:

1. Maximize the customers who flow through the system, while
2. Observing this flow for signs of network failure such as queue length, and
3. Repairing failures as quickly as possible.

In order to develop this model, it was necessary to become familiar with the simulation. Once this was accomplished, the next step was to formalize procedures to be employed in the model in such a way that it would be feasible to write a computer code. The procedures had to be general, but at the same time, they had to be effective in controlling and operating the system. The model was named KARLA as a follow up of a previous research effort of the Center for Man-Machine Systems Research, namely KARL. A flow chart of KARLA is depicted on Figure 1.

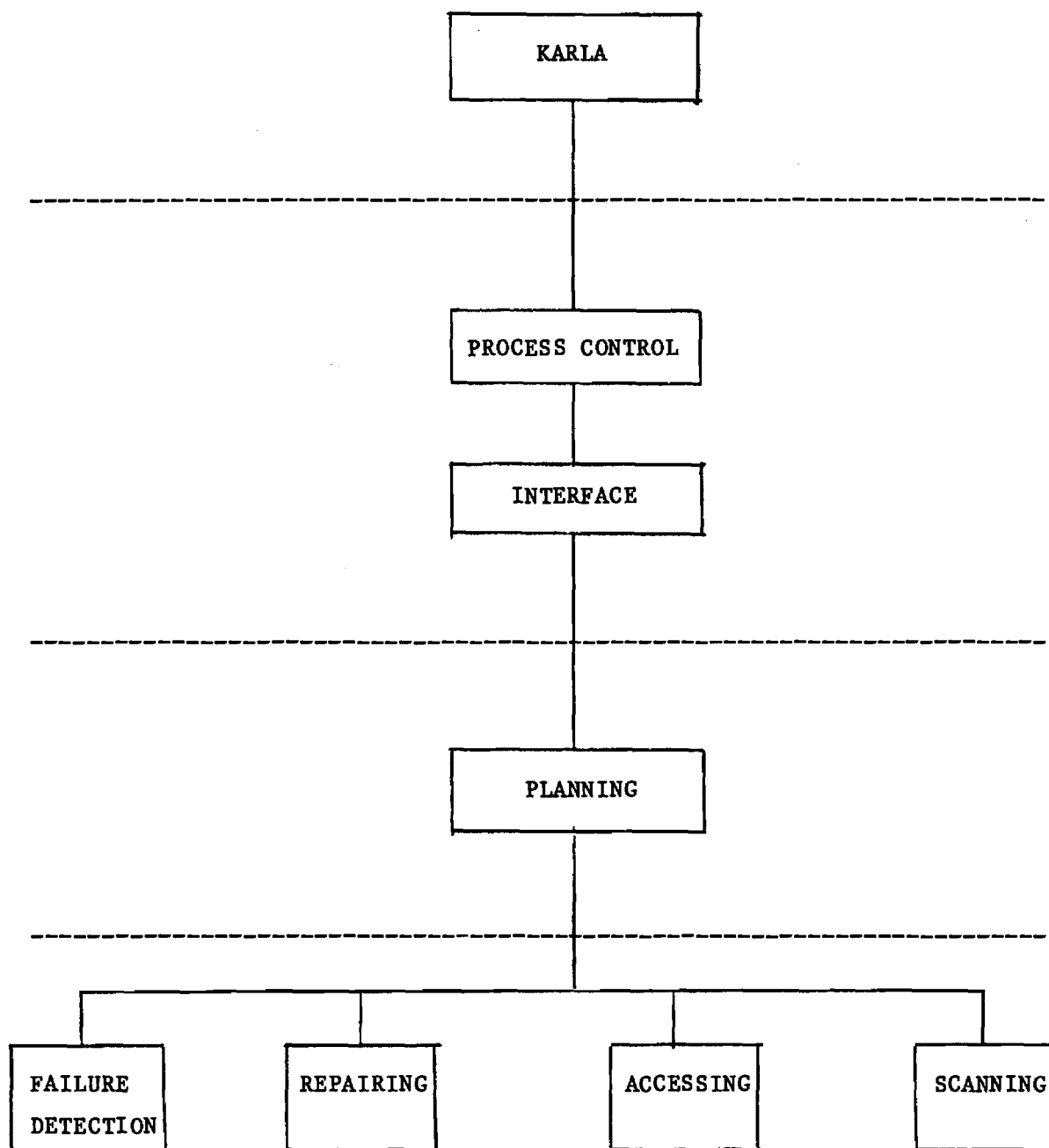


Figure 1. Flow-chart of KARLA

The model consists of a series of procedures that mimic human actions. Since the ultimate goal of a rule-based model is to duplicate the behavior of the human being modeled, rather than concentrating on final results (Hunt, 1981), special care was taken to include rules that duplicate observed human behavior.

#### PSYCHOLOGICAL ISSUES

There are a number of methodologies that could be used to design the model. The approach chosen was to use a rule-based model with features typical of human information processing. Alternative approaches might be a rule-based expert system approach, which would be similar except that it would try to outperform humans. A decision-theoretic or control-theoretic model could also be developed. For this reason, the main issues from a psychological perspective that were implemented in the model are sequences of commands, short term memory limits, optimal repair behavior, scanning, revisit inhibit, and reaction time delay.

Commands issued as sequences: Minsky (1975), Schank and Abelson (1977), and Rich (1977) have developed several models to show how people trade off between storage and computation. Basically, humans represent knowledge not in purely canonical or literal form, but with a certain tradeoff between storage and computation. The model acts similarly. On one side of the spectrum, it could employ no memory and just issue one action at a time. This is the way KARL (Knaeuper, 1983) does it. On the other hand, the model could be comprised of different scripts. Each script could represent whole sets of strategies appropriate for different situations. Humans are not likely to represent knowledge either way. Rasmussen and Jensen (1974) argue that a troubleshooter must have some sort of a mental model of the environment being monitored. Studying the subjects' performance, it became clear that operators coupled their knowledge of the model with their increasing expertise to structure short series of commands. These series of commands are situation-dependent, i.e., goals trigger the use of familiar scripts which in turn cause the execution of an array of actions (Schank and Abelson, 1977). This coding scheme has been called "chunking" (Miller, 1956). For instance, when the queue length in a node becomes too large to the point that it almost

reaches the capacity of the node, both the subjects and KARLA issue a "down" command followed by a "test" command.

Short-term memory limit: Due to STM limitations, troubleshooters have a maximum number of actions that they can incorporate in their procedures. Baddeley and Hitch (1974) conclude that the more slots that are filled in an operator's memory, the less working space is available for problem solving and other calculations. Reasonable estimates range from five to nine items. Due to the dynamic nature of the task, a maximum of four commands per script was viewed as realistic. This is the strategy that the model incorporates in its knowledge base. For instance, after the model has finished repairing a node, its "memory" of commands is blank and must look for familiar situations that would trigger rules and fill its memory of commands.

Optimal repair behavior: Not all failures have the same importance in the environment that KARLA controls. When simultaneous failures occur, the model will look for those that are more important in the system, and compensate for them first; once this is accomplished, it will take care of the failures with lesser importance. Failure compensation, thus, can be done in two levels. The first level compensates for a failure as it occurs (Knaeuper, 1983). The second level maximizes the failure compensation according to the conditions of the system. "Maximizing compensation" means that failures are repaired following a hierarchy of importance. This is what distinguishes a good from a bad operator: the former will not only do his job, but will try to do it the best possible way; the latter will just "get it over with." KARLA takes the "good" operator approach to the failure compensation issue (Henneman and Rouse, 1984).

Scanning: Operators were observed to seek out failures, even when there was no apparent cause to suspect one, rather than just wait and respond when there was a reason to suspect a failure. This was particularly true in the low failure rate systems. The model incorporates a similar strategy. During periods of little or no action it scans the system looking for failures.

Revisit inhibit: This simply means that the model will not re-examine a node that is known to be without failures because it was recently tested. This is actually another form of short term memory. It was incorporated in the model because subjects were known to behave this way. In a research effort, Platz, Rasmussen and Skanborg (1975) concluded that human response to alarms were faster if the operator could have had some prior knowledge of the failure. In revisit inhibit, the model knows there is not a failure in a specific node and can concentrate its troubleshooting on the other nodes.

Reaction time: The model's response to the inputs is not immediate. Rather, there is a lag between the moment an alarm occurs and the instance when the model tries to stabilize the system. Rouse (1980) observes that a simple model mapping a single input to a single output is inadequate because of two main reasons. First, humans require a finite amount of time to react to stimuli, and, secondly, human's neuromotor system prohibits the instantaneous movement of limbs. KARLA includes this consideration and defers its reaction to alarms mimicking neuromuscular lag and reaction time.

## RESULTS

Two kinds of performance were assessed. The first one, called "open-loop," involved letting the model run by itself, just like a regular operator, and obtaining the production measures at the end of the run. The second one, termed "closed-loop," involved behavioral comparisons. In other words, subjects' data files and the model run in parallel with the subjects controlling the scenario. The commands issued by both were recorded and later compared to determine whether or not the subjects and the model issued the same command when they were controlling the same situation.

### Open-loop Run

To compare the model with the subjects on total performance, KARLA was allowed to run by itself, in an open loop manner without any knowledge of the subjects' actions. Seven different measures of production performance were used: number of customers left in the system, average time in the system, number of customers lost, number of customers denied service, total number of customers served, and number of failures repaired.

Table 1 shows the number of customers left in the system for both the subjects and KARLA under the different experimental conditions. The number of customers left in the system seems to be affected by the failure rate, number of nodes, and number of levels. KARLA's performance seemed to be the most affected in a 16 node system. The differences between the subjects and KARLA was small.

Average time spent in the system is depicted in Table 2. The model performed consistently better than the subjects. Only in the high failure rate group of the 16-nodes, three-level systems was the average of the subjects better than the model's performance. As the systems started increasing in size, the times also started increasing, this is only natural since it takes a longer time to be processed in a larger system. The main reason behind customers being delayed in the system is the number of failures in the simulation. KARLA's performance can be explained from this point of view. Simply put, the model's strategies caused less failures in the system, therefore, the average travel time of a call was reduced.

### Closed-Loop Run

In the closed-loop run, the scenario was controlled by the subject's actions. Given a particular situation, the subject and the model decided what the appropriate action should be and both actions were recorded, but it was actually the subject's command that was fed back to MABEL and, in turn, determined the next state.



At the end of a run, matches between the subject's commands and KARLA's commands were established. Basically, three types of match were considered. The first type of match is the one when the subject and the model gave the same command at the same time. For instance, both KARLA and the subject issued 't', or 'm' at the same time. The second type of matching relates to the fact that, according to Henneman (1982), there are four types of command: access commands (d, u), monitor commands (m, s, c), diagnostic commands (t, n), and, control commands (r, l). Commands within each category try to accomplish a common goal, i.e., access, control, etc., therefore, when the subject issued a command, and, KARLA issued a command from that same group, a match was recorded.

The third matching sequence measure was similar to the second except that out-of-sequence agreement was allowed. In some instances, a subject would issue a series of commands with a specific purpose in mind, e.g., to find out if there is a failure in a node one level below, the subject types: 'u', 'm', 't'. At the same time, the model would try to accomplish the same purpose, but the strategy could be different, e.g., 'u', 't', 'm'. In this case, the objective is the same but the sequence of commands differs.

The performance of KARLA on the three levels of matching is as follows:

LEVEL	PERCENTAGE EQUIVALENT COMMANDS
1	.47085
2	.01015
3	.40661

Since the percentage matching measures are independent from each other, the overall percentage matching is 88.76 %.

Table 1. Number of Customers

SYSTEM	SUBJECTS AVERAGE	KARLA	AVG. PERCENTAGE DIFFERENCE
222C	18.83	20	- 6.21 %
222D	16.83	22	-30.72 %
223C	45.50	18	60.44 %
223D	33.83	36	- 6.41 %
332C	72.17	59	18.25 %
332D	82.50	74	10.30 %
333C	139.50	116	16.85 %
333D	131.17	176	-34.18 %
442C	230.33	204	11.43 %
442D	222.50	240	- 7.87 %
443C	527.00	500	5.12 %
443D	523.50	577	-10.12 %
			-----
			Avg. 2.24 %

Table 2. Average Time in the System

SYSTEM	SUBJECTS AVERAGE	KARLA	AVG. PERCENTAGE DIFFERENCE
222C	2.28	1.84	19.30 %
222D	2.90	2.42	16.55 %
223C	5.36	2.01	6.25 %
223D	4.76	4.54	-0.23 %
332C	7.96	7.63	4.15 %
332D	8.68	8.35	3.80 %
333C	14.44	13.49	6.58 %
333D	15.51	13.99	9.80 %
442C	25.33	25.16	0.67 %
442D	26.79	26.30	1.83 %
443C	63.91	63.78	0.20 %
443D	64.00	64.17	-0.27 %
			-----
			Avg. 5.72 %

## REFERENCES

1. Baddeley, A.D. and Hitch, G. Working Memory. In G.H. Bower (Ed.) The Psychology of Learning and Motivation, Vol. 8. New York: Academic Press, 1974.
2. Henneman, R.L. and Rouse, W.B. Human Performance in Monitoring and Controlling Hierarchical Large Scale Systems. IEEE Transactions on Systems, Man, and Cybernetics. Vol. SMC-14, No. 2, March/April 1984.
3. Hunt, R.M. Human Pattern Recognition and Information Seeking in Simulated Fault Diagnosis Tasks. Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1981.
4. Knaeuper, A. A Model of Human Problem Solving in Dynamic Environments. Master's Thesis, Georgia Institute of Technology, Atlanta, 1983.
5. Miller, G.A. The Magical Number Seven, Plus or Minus Two. Psychological Review, 1956, 63, pp. 81-97.
6. Minsky, M. A Framework for Representing Knowledge. In P.H. Winston (Ed.) The Psychology of Computer Vision. New York: McGraw-Hill, 1975.
7. Platz, O., Rasmussen, J. and Skanborg, P.Z. Response Times of Operators in a Control Room. Roskilde, Denmark: Riso National Laboratory, 1975.
8. Pylyshyn, Z. Representation of Non-Linguistic Forms. Proceedings of Theoretical Issues in Natural Language, Proceedings (TINLAP), Cambridge, MA, 1975.
9. Rasmussen, J. Models of Mental Strategies in Process Plant Diagnosis. In J. Rasmussen and W.B. Rouse (Eds.) Human Detection and Diagnosis of System Failures, Plenum Press, New York, 1981, pp. 241-258.

10. Rasmussen, J. and Jensen, A. Mental Procedures in Real Life Tasks: A Case Study of Electronic Troubleshooting. Ergonomics, 1974, 17(3), pp. 293-307.
11. Rich, E. User Modeling via Stereotypes. Cognitive Science, 1979, 3, pp. 329-354.
12. Rouse, W.B. Systems Engineering Models of Human Machine Interaction, North Holland, New York, 1980.
13. Schank, R.C. and Abelson R.P. Scripts, Plans, Goals, and Understanding. Hillsdale, New Jersey, Lawrence Erlbaum, 1977.

CENTER FOR MAN-MACHINE SYSTEMS RESEARCH

SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING

ATLANTA, GEORGIA 30332

(404) 894-3996

*E-24-655/Rouse  
Joye*

May 4, 1984

Dr. Marshall Narva  
Army Research Institute  
5001 Eisenhower Avenue  
Alexandria, VA 22333

Dear Dr. Narva:

Enclosed please find four copies of our bimonthly report for the period February 1, 1984-March 31, 1984.

Sincerely,

William B. Rouse  
Professor and Director

WBR:paj

cc: L.H. Bowman  
F. Cochran

Enclosures

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Bimonthly Report

For the Period February 1, 1984 - March 31, 1984

Contract MDA 903-82-C-0145  
(June 1, 1982 - July 1, 1985)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404/894-3996)

During the February-March reporting period, Dick Henneman completed development of CAIN, the contextually-augmented version of MABEL. The first formal experiment with CAIN was performed involving eight subjects, each of whom performed 13 sessions of approximately one hour in length each. Independent variables included number of levels in the display heirarchy and degree of interconnectivity among nodes. Analysis of the wealth of data produced is focusing on evaluating measures of large-scale system complexity as well as comparisons with the results of the earlier experiment with MABEL. The results of these two experiments, particularly in terms of complexity of problem solving, will be the basis of Dick's Ph.D. thesis, which should be completed in the next two or three months.

The initial model of human problem solving with MABEL has been programmed and is being evaluated by Eduardo Viteri. This work will comprise Eduardo's M.S. thesis, which should be completed this summer.

As of March 31, 1984, approximately \$153,000 has been spent.



HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Bimonthly Report

For the Period June 1, 1984 - July 31, 1984

Contract MDA 903-82-C-0145  
(June 1, 1982 - July 1, 1985)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404/894-3996)

Work has continued relative to the development and assessment of measures of large scale system complexity. Time series analysis is being used to determine transfer functions relating two measures of complexity, namely structural and strategic complexity, to the average time to find failures within the system. These efforts should be completed within the very near future, and the results will be reported in the PhD thesis of Richard Henneman.

In addition, a detailed analysis of subjects' strategies in operating the context-specific simulation was performed. Results suggest that subjects achieve better performance by issuing more monitoring commands rather than commands related to moving among display pages. In other words, the better subjects achieve higher levels of performance by using the more powerful monitoring commands to gain more information about the system than could efficiently be gained by actually branching among display pages.

As of July 31, 1984, approximately \$187,000 has been spent.

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Bimonthly Report

For the Period August 1, 1984 - September 30, 1984

Contract MDA 903-82-C-0145  
(June 1, 1982 - July 1, 1985)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404/894-3996)

Work over the past two months has concentrated on further investigating the nature of complexity in large scale systems, specifically, the simulated communication network (CAIN) described in a previous progress report. It was hypothesized that the complexity of the system (as reflected by the difficulty that people have in finding faults in the system) is shaped by two things: the structure of the system and the human's understanding of the system as reflected by his strategy. From this perspective, the complexity of a system changes with time.

Due to the dynamic nature of complexity, time series analysis was identified as the appropriate methodological tool with which to analyze the data. Two input (i.e., the two complexity measures) transfer function models were developed for each subject to predict the average time needed to locate failures within the system. The approach was quite successful in that no structure remained in the autocorrelation function of the residuals and the models consistently explained more than 80% of the variance within the original data.

A problem arose in trying to find a consistent interpretation of the transfer functions. The equations contained terms at time lags ranging from a few seconds to a few minutes. In addition, there appeared to be little consistency among subjects in terms of the structure of the equations.

A reasonable explanation for this high degree of variability lies in identifying several different modes of failure identification (e.g., topographic, symptomatic, and serendipitous) and also several different types of event associated with the identification of a failure. For example, it can be shown that the complexity of the system at the time a failure occurs can increase the time needed to locate that failure; on

the other hand, the complexity of the system at the time a symptom of that failure first occurs can decrease the time needed to locate the failure. There appears to exist a relationship between these inter-event times and the lags present in the transfer functions.

More detail relative to these results may be found in the forthcoming Ph.D thesis of Richard Henneman. This work should be completed within the next several weeks.

As of September 30, 1984, approximately \$195,000 had been spent.

October through December has been the "home stretch" for both Dick Henneman and Eduardo Viteri. Dick's Ph.D. thesis on "Human Problem Solving in Complex Hierarchical Large Scale Systems" was completed in mid-December and he will give his oral defense in early January. Eduardo's M.S. thesis on "A Rule-Based Model of a Human Operator Controlling a Complex Communication Network" was also completed in mid-December. Copies of the Center technical reports based on these theses will be forwarded to ARI in early 1985.

As of November 30, 1984, approximately \$222,000 had been spent.

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Bimonthly Report

For the Period December 1, 1984 - January 31, 1985.

Contract MDA 903-82-C-0145  
(June 1, 1982 - July 1, 1985)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404-894-3996)

Work over the past two months has been focused on the interpretation of results from the second experiment from the perspective of measuring system complexity. Transfer functions were developed that related measures of structural and strategic complexity to average time to failure diagnosis. The analysis suggested that important factors affecting the structure of these equations (and hence, failure diagnosis time) were system related (e.g., the speed with which failure symptoms propagate through the system) and strategy related (e.g., whether subjects rely on a symptomatic, topographic, or serendipitous search strategy.)

Results also emphasized the different implications that complexity may have for normal system operation and human failure diagnosis performance. Although certain system characteristics (such as multiple levels and high degree of interconnectedness) may help to avoid the short term effects of failures, these same characteristics may have the dual effect of making the human supervisory controller's task more difficult.

These findings are reported in the Ph.D. thesis of Dick Henneman. The thesis, which was completed in January, will be forthcoming as a technical report. Eduardo Viteri's Master's thesis (the development and analysis of a rule-based model that controls the simulated communication network) will also appear as a technical report. The abstracts of these theses are attached.

As of January 31, 1985, approximately \$240,000 had been spent.



A RULE-BASED MODEL OF A HUMAN OPERATOR IN A  
COMPLEX COMMUNICATION NETWORK

Eduardo T. Viteri

51 Pages

Directed by Dr. John Hammer

Since the beginning of the Industrial Revolution, the time humans have spent in manual activities has decreased drastically and, on the other hand, more and more time has been spent in monitoring automated processes and solving occasional problems caused by the machines that have supplanted the manual laborers. One of the environments where automation is playing an increasingly crucial role is the communications field. This thesis is concerned with the development of a rule-based model of human behavior in fault diagnosis of a large scale hierarchical communication system. The model has been named KARLA as a follow up of a model developed by Knaeuper and Rouse (1983).

The predecessor of the model developed in this thesis is presented. Then, the simulation that KARLA controls is explained. Next, the structure of KARLA is explained. Results are presented from applying this model to modeling a human operator in a complex communication network.

---

John Hammer, Advisor

Human Problem Solving in Complex  
Hierarchical Large Scale Systems

Richard L. Henneman

240 Pages

Directed by Dr. William B. Rouse

Humans supervising highly automated, hierarchical, large scale dynamic systems, such as a communications network, must often take control action during unforeseen failure or emergency situations. These unexpected events combine with the system size and structure to produce possibly complex tasks for humans. This thesis explores human performance in monitoring and controlling these complex systems.

Two experiments used versions of a computer simulated communication network. Subjects monitored and controlled the system via a video display and keyboard. Subjects were told to optimize system performance (e.g., maximize number of customers served and minimize customer processing time) by diagnosing failed components and managing network resources (e.g., by shedding load).

The first experiment used a relatively context-free representation of a communication network in an experiment that varied the number of system levels, cluster size (number of nodes/display page), and node failure rate. Subject performance degraded with increasing number of levels and decreasing cluster size. Failure rate only affected subjects' strategy. The unexpected result concerning cluster size was suggested to be due to the number of connections between nodes.

The second experiment used a contextually augmented version of the original simulation. Experimental variables were number of connections between nodes and number of levels. Cluster size remained constant. Results supported those from Experiment One: increasing number of levels and decreasing the connectivity between nodes degraded performance.

The second experiment also investigated the nature of complexity in a large scale system. Two dimensions (and associated measures) of complexity were proposed: complexity due to system structure and complexity due to human strategy. Transfer functions relating the two complexity measures to average time to failure diagnosis were developed. Results indicated that the distinction between structural complexity and strategic complexity is appropriate.

Results also emphasized the different implications that complexity may have for normal system operation and human failure diagnosis performance. Although certain system characteristics (such as multiple levels and high redundancy) may help to avoid the short term effects of failure, these same characteristics may have the dual effect of making the human supervisory controller's task more difficult.

---

William B. Rouse, Advisor

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Bimonthly Report

For the Period February 1, 1985 - March 31, 1985

Contract MDA 903-82-C-0145  
(June 1, 1982 - July 1, 1985)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404-894-3996)

Conceptual work on a model of human performance in CAIN was initiated. The problem solving model of Rouse [1983] was used as a starting point; however, due to several features of the CAIN environment, the original model must be augmented. For example, since multiple failures can exist in CAIN, the human operator may have several different tasks to perform at any one time. Moreover, since the states of the system evolve with time, the relative importance of these tasks will also change. In light of these domain characteristics, the proposed model is characterized by a number of "low-level" tasks (e.g., recognition and classification, planning, and execution) that are prioritized according to a higher level planning function.

Besides task prioritization, the role of contextual knowledge is of importance to the performance of this control task. The proposed model, therefore, incorporates two types of knowledge: knowledge about the system structure and context, and knowledge about how to do things (i.e., procedural knowledge). Rob Andes, a new masters degree student, has started reviewing the literature relative to the role of contextual knowledge in human problem solving.

The goal of this modeling effort is to support an on-line aid or "coach" to human operators performing the task. A robust, flexible model of human performance in this task should be able to support operators using any of a variety of strategies and functioning at differing levels of expertise.

As of March 31, 1985, approximately \$272,000 had been spent.

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse  
Richard L. Henneman

Annual Interim Report

For the Period June 1, 1984 - May 31, 1985

Contract MDA 903-82-C-0145  
(June 1, 1982 - June 1, 1986)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404-894-3996)

## INTRODUCTION

Current trends in computer and communications technology are leading to the development of many highly integrated systems in the domains of communications, transportation, manufacturing, etc. Most of these systems can be represented as large networks of nodes and arcs where nodes denote people, destinations, or machines and arcs denote communication lines, transportation routes, or a variety of activities. Because these systems are highly integrated, it is not unusual for there to be hundreds or thousands of nodes and arcs. Networks of this size and level of connectivity are very complex systems.

Complexity is further increased by the dynamic nature of these networks. The states of the nodes and arcs (i.e., levels, flows, etc.) usually evolve in time and are not amenable to instantaneous control. Further, the demands placed upon the networks are often time-varying, with occurrences of peak demands not always being predictable.

This program of research is concerned with the problem solving behavior of the human whose role is network controller or operator. The job of the network controller is to manage the assets of the network (i.e., nodes and arcs) so as to maximize network efficiency. Further, during peak demand periods, the controller may have to implement control procedures such as load shedding and priority scheduling to assure that overloads do not degrade network performance.

For many aspects of this job, the network controller has computer aids or, in fact, may simply have to monitor an automated system which performs many of the above functions. However, system failures or unusual environmental demands can require that the human intervene and manually control the network. The human's abilities to solve these types

of problem are not well understood. In fact, human problem solving in complex dynamic environments is an area where few research results are available. This area is the topic of the research program whose progress is reported here.

## PROGRESS

This section briefly summarizes progress during the first three years of this four-year program of research. Considerably more detail about the most recent results can be found in the papers included in the Appendix.

Most of the first year was devoted to developing an experimental scenario and evaluating the impact of its parameters on human problem solving performance [Henneman and Rouse, 1984a]. Communications networks were chosen as the experimental context. After reviewing a variety of documentation on human control tasks in both commercial and military communications networks, an experimental scenario called MABEL was designed and programmed. MABEL requires subjects to monitor a large-scale automated communications network via a hierarchical multi-page CRT display. Much as discussed in the Introduction, subjects have to manage network assets and, in the event of a failure, intervene to diagnose the failure, compensate for its impact, and restore normal operation.

For the first formal experiment with MABEL, the effects of three independent variables were studied: 1) number of nodes per display, 2) number of levels in the display hierarchy, and 3) failure rate per node. Twelve subjects each participated in six experimental sessions. Overall, this initial experiment with MABEL produced two results of particular interest. First, the effects of number of levels in the hierarchy were



often very strong, producing up to a five-fold degradation of performance for a modest change from two to three levels. The second result of note is that rather different strategies seemed best for different combinations of independent variables. This leads to the question of whether humans can be trained to adapt appropriately or if some form of aided adaptation is needed.

The second year of this research involved two efforts. One effort concerned the development of a rule-based model of human problem solving in the MABEL environment [Viteri 1984]. One general impression that emerged from the experiment and the modeling efforts was that MABEL lacked the contextual richness necessary to provide the type of problem solving environment required for this research. Perhaps the best indication of this is the simplicity of Viteri's model even though it compares fairly well with subjects' behavior.

This observation led to a decision to enhance substantially the contextual aspects of MABEL. The second formal experiment [Henneman and Rouse 1984b, 1985, Henneman 1985] used a contextually augmented version of MABEL called CAIN (Contextually Augmented Integrated Network). The scenario contained cues and associative links (e.g., non-varying geographic node names, recurring failures, and non-uniform loading) to produce a higher fidelity simulation. Cluster size was kept constant at 16 so that subjects could learn and recall context-dependent aspects of the system. Experimental variables were number of connections between nodes (high,low) and number of levels (2,3). Eight subjects each participated in thirteen experimental sessions. Results supported those from Experiment One: increasing number of levels degraded performance, as did decreasing the connectivity between nodes.

Efforts in the third year of this research have been directed towards realizing a major objective of the second experiment, namely, to investigate the nature of complexity in a large scale system [Henneman and Rouse 1985, Henneman 1985]. Two dimensions (and associated measures) of complexity were proposed: complexity due to the structure of the system and complexity due to the strategy of the person trying to control the system. Complexity was considered to be a dynamic property of a human-machine system. Complexity is time-dependent and multi-dimensional; thus, time series analysis was used to develop transfer functions relating the two complexity measures to average time to failure diagnosis. Results indicated that the distinction between structural complexity and strategic complexity is appropriate.

Results also emphasized the different implications that complexity may have for normal system operation and human failure diagnosis performance. A very complex system may function quite well under normal operating conditions. The system is able to absorb the effects of failures to a certain extent while maintaining an adequate level of performance. However, when the problem becomes so critical that the human monitor must intervene and find the problem, the task of failure diagnosis may be very difficult. In summary, although certain system design characteristics may help to avoid the short term effects of failures, these same characteristics may have the dual effect of making the human supervisory controller's task more difficult. These results are presented in the paper included in the Appendix [Henneman and Rouse 1985].

Other efforts in the third year of this research have been directed towards the conceptual development of a sophisticated model-based

performance aid for humans monitoring and controlling CAIN. The proposed rule-based model is described in a paper in the Appendix [Henneman 1985b]. The model is characterized by three stages of problem solving (recognition/classification, planning, and execution) that are prioritized according to the model's knowledge about the task and about the system (e.g., contextual relationships among components).

#### FUTURE WORK

On-line implementation of the model proposed in the paper in the Appendix has just started. Future plans include the experimental evaluation of the model as an on-line performance aid. The proposed experiment should compare the task performance of two groups of subjects, one of which performs without the aid and the other with the aid. An interesting side issue to explore involves the representation and use of the contextual information included in the experimental scenario.

## REFERENCES

1. Henneman, R.L. and Rouse, W.B. "Human problem solving in large-scale networks," Proceedings of the 1982 International Large-Scale Systems Symposium, Virginia Beach, October 1982, pp. 293-297.
2. Henneman, R.L. and Rouse, W.B. "Human performance in monitoring and controlling hierarchical large-scale systems," Proceedings of the 27th Annual Meeting of the Human Factors Society, Norfolk, Virginia, October 1983, pp. 685-689.
3. Henneman, R.L. and Rouse, W.B. "Human performance in monitoring and controlling hierarchical large-scale systems," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-14, No. 2, pp. 184-191, March/April 1984a.
4. Henneman, R.L. and Rouse, W.B. "Assessing the complexity of a large-scale system: measures of system structure and human strategy," Proceedings of the 1984 IEEE International Conference on Systems, Man, and Cybernetics, Halifax, Nova Scotia, October 1984b.
5. Viteri, E. "A rule-based model of a human operator in a complex communication network," MSIE Thesis, Georgia Institute of Technology, 1984.
6. Henneman, R.L. "Human problem solving in complex, hierarchical large-scale systems," Ph.D. Thesis, Georgia Institute of Technology, 1985a.
7. Henneman, R.L. "A model of human performance in a large scale dynamic system," Proceedings of the 1985 IEEE International Conference on Systems, Man, and Cybernetics, Tucson, AZ, November 1985b.
8. Henneman, R.L. and Rouse, W.B. "On measuring the complexity of monitoring and controlling large scale systems," submitted for publication, 1985.

## APPENDIX

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Bimonthly Report

For the Period June 1, 1985 - July 31, 1985

Contract MDA 903-82-C-0145  
(June 1, 1982 - July 1, 1985)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404-894-3996)

Work during this period has been directed towards refining and implementing the model based aid for human performance in monitoring and controlling CAIN. In particular, theoretical consideration has been given to how knowledge should be represented in each of the model components (i.e., prioritization, recognition/classification, planning, execution, and contextual knowledge). Also, efforts have been made towards determining means of structuring these knowledge representations in a computer program.

The VAX 11/780 of the Center for Man-Machine Systems Research has recently changed to a UNIX operating system; thus, the model will be programmed in C. In addition, due to the change in operating system, modifications were made to the CAIN Pascal program in order to make it UNIX compatible.

As of July 31, 1985, approximately \$317,000 had been spent.

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Bimonthly Report

For the Period August 1, 1985 - September 30, 1985

Contract MDA 903-82-C-0145  
(June 1, 1982 - July 1, 1986)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404) 894-3996



Work during this period has continued the development of the model of human performance in monitoring and controlling the CAIN system. Efforts have focused on the identification of rules that adequately describe aspects of task performance. In this task, the decision of what rule to apply is, in general, not difficult. A much more difficult decision (and, hence, an aspect of performance more difficult to model) is when to apply a particular rule. Thus, additional efforts have focused on the prioritization and timing of rules that apply to various task situations. Recent ideas have considered a fuzzy prioritization process, perhaps based on the notions of complexity and uncertainty used in Henneman's thesis.

Two personnel changes have occurred during this period. Rob Andes left school to take a job. Klaus Zinser, a new M.S. student from Germany, has started to work on the project.

As of September 30, 1985, approximately \$337,000 had been spent.

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse

Bimonthly Report

For the Period October 1, 1985 - November 30, 1985

Contract MDA 903-82-C-0145  
(June 1, 1982 - May 31, 1986)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, GA 30332  
(404-894-3996)

### Current Work

Work during this period has concentrated on implementing the model of human performance in CAIN. The general form of the model has been presented in an earlier report; some of the specific implementation details that are currently being incorporated into the model include a frame representation of system (contextual) knowledge and a fuzzy decision making process that prioritizes subtasks. In addition, CAIN is being moved to run on an AT&T 3B2 computer with enhanced graphics capabilities. AT&T made a gift of two of these computers to the Center for Man-Machine Systems Research. By running the CAIN system and the human performance model in parallel on separate machines, there will be no competition for computer resources from either of the two programs or other users. This competition has been a problem in previous experiments.

### Work Remaining

Given that six months remain on this contract, the rest of this report will present a list of remaining tasks, and the major issues that will be considered in the final phase of this research program. Considering the tasks remaining first, the following list describes each task, gives the approximate date of completion, and indicates who will be responsible for the task (RLH = Henneman, KZ = Zinser, WBR = Rouse).

1. Transfer CAIN to AT&T 3B2/Dec 1985 (RLH)
2. Complete programming of model/Jan 1986 (KZ)
3. Collect subject data to compare model performance/Jan 1986 (KZ,RLH)
4. Implementation of model-based aid/Feb-Mar 1986 (KZ,RLH)
5. Run experiment comparing aided vs. unaided subjects/April 1986 (KZ)
6. Data analysis and interpretation/May 1986 (KZ,RLH)
7. Final report (RLH,WBR)

### Theoretical Considerations

These tasks are to be conducted in light of the theoretical issue of human knowledge representation. We are concerned with how people represent knowledge and how that representation changes with time. More specifically, we are interested in knowledge representation in a complex task environment, an environment in which the system state is dynamic and decisions are often made on the basis of incomplete knowledge. Thus, a related (but higher level) issue involves decision making in a complex task: how does both the representation of knowledge and aspects of the task environment affect human decision making in a complex environment?

The current modeling efforts should help in resolving these issues. One potentially useful idea under consideration is to compare the performance of a model that makes full use of its context-dependent knowledge structures to that of a model stripped of its contextual knowledge. Differences in model performance patterns should give insight to aspects of human performance that are independent of the context of the system.

In addition, an increased understanding of human knowledge representation and decision making should suggest ways to aid human performance in complex environments. The means of aiding human performance in this task will rely on an on-line model of a human operator. The model is such that it will contain knowledge structures and performance mechanisms consistent with that of a human operator. Thus, the model should be able to provide aid consistent with the human operator's needs.

### Practical Considerations

The implementation of such a model based performance aid raises several important practical issues. These issues can be broadly grouped into three

main categories. First, decisions relating to what information should be provided must be made. For example, the aid may be capable of presenting several different types of information (e.g., procedural and contextual). What information will be most useful to the human operator? In order for the advice to be useful, it must not be superfluous to the operator's current state of knowledge. The model basis for the aid should provide guidance relative to the operator's current state of knowledge.

Second, decisions relating to when advice should be provided must be made. In a dynamic environment, the system state is constantly changing. Relevant advice at one point in time may be irrelevant at a later time. Thus, some very important questions ask when information should be presented, and, perhaps just as important, when should information stop being presented?

Finally, third, how the information is presented, or the mode of presentation, must be selected. The resolution of this issue is probably more technology driven than the other two issues, but there are some interesting questions that can be raised relative to mode of display (e.g., visual vs. auditory) and integration of the advice with the actual system display.

Other questions combine elements of all these three issues. For example, if the aid provides advice and the human performs an action completely different from the advice, how should the aid respond? It is important to note that the resolution of these issues should have major implications for not only the simulated system under consideration here, but for any real life system that is to provide intelligent aid to a human operator.

## Conclusion

To conclude, the ideas presented in this report represent a refinement of the originally proposed work. The emphasis has shifted to address how people represent and use knowledge in controlling a very large and complex system. Means of using this understanding will be explored in order to aid human operators.

As of November 30, 1985, approximately \$369,000 had been spent.

## HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

MDA 903-82-C-0145

Georgia Institute of Technology

1 June 1982 - 31 May 1986

William B. Rouse

**SCIENTIFIC OBJECTIVES:** The primary objectives of this project are to understand and support human performance in the task of monitoring and controlling a large dynamic system, such as a communication network or a command and control network. Emphasis is placed on both human-system design requirements and human performance modeling.

**APPROACH:** An experimental scenario has been developed that involves monitoring and control of a large-scale, dynamic network that is very similar to the current telephone system in the United States. Network information is displayed to subjects in a hierarchical, multi-page manner that is similar to display systems found in many large-scale systems. Independent variables that can be manipulated include: 1) number of nodes per display page, 2) number of levels of pages in the hierarchy, 3) failure rate per node, and 4) level of network redundancy. Subjects are instructed to control the network such that all demands are satisfied while avoiding inordinate processing delays. In addition, they must diagnose and correct node failures that lead to losses of resources that could eventually compromise the overall performance objectives of satisfying demands and avoiding delays.

**RESULTS AND CONCLUSIONS:** Thus far, two experiments have been performed. The first experiment involved 12 subjects and considered the effects of nodes per display page, number of levels of pages, and failure rate per node. Number of levels of pages was found to have the strongest effect, producing up to a five-fold degradation of performance for a modest change from two to three levels. The second experiment involved 8 subjects and considered the effects of number of levels of pages and level of network redundancy. The primary purpose of this experiment was to evaluate two measures of problem solving complexity. The first measure is dependent upon the structure of the system; the second measure is dependent on the strategy of the person controlling the system. Results suggest that this distinction is appropriate. In addition, results emphasize the different implications that complexity can have for normal system operation and human failure diagnosis performance. Although system design characteristics such as redundancy may help to avoid the short term effects of failures, these same characteristics may have the dual effect of making the human supervisory controller's task more difficult.

Recent efforts have been directed towards developing a model-based performance aid for people controlling large scale systems. The model consists of four main components: task prioritization, recognition/classification of failure situations, planning, and task execution and monitoring. In addition, the model contains an explicit representation of the contextual knowledge needed to control the system effectively. The model will operate on-line, while assisting people in such activities as prioritizing tasks and recalling contextual knowledge about the system structure. At least one more experiment is planned to evaluate the effectiveness of this aiding approach.

POTENTIAL APPLICATIONS: Applications of results from this project are two fold. First, results from the empirical experiments and theoretical investigation of complexity have direct relevance to the design of large scale human-machine systems. Second, the demonstration of the viability of the model-based performance aiding approach should suggest a useful means of improving human-system performance in a variety of complex environments.

#### REPORTS:

1. "A Model of Human Performance in a Large Scale System," R.L. Henneman, Proceedings International Conference on Cybernetics and Society, November 1985, Tucson, AZ.
2. "Human Problem Solving in Large Scale Dynamic Networks," R.L. Henneman. Ph.D. Dissertation, Georgia Institute of Technology, CMMSR Report No. 85-1, January 1985.
3. "A Rule-Based Model of a Human Operator in a Complex Communication Network," Eduardo T. Viteri. M.S.I.E. Thesis, Georgia Institute of Technology, CMMSR Report No. 84-3, December 1984.
4. "Assessing the Complexity of a Large Scale System: Measures of System Structure and Human Strategy," R.L. Henneman and W.B. Rouse, Proceedings International Conference on Cybernetics and Society, October 1984, Halifax, Nova Scotia.
5. "Human Performance in Monitoring and Controlling Hierarchical Large Scale Systems," R.L. Henneman and W.B. Rouse, Proceedings of the Human Factors Society 27th Annual Meeting, Norfolk, VA, October 1983.
6. "Human Problem Solving in Large Scale Networks," R.L. Henneman and W.B. Rouse, Proceedings of the IEEE International Large Scale Systems Symposium, Oct. 1982, pp. 293-297.

#### ARCHIVAL PUBLICATIONS:

1. "Human Performance in Monitoring and Controlling Hierarchical Large Scale Systems," R.L. Henneman and W.B. Rouse, IEEE Transactions on Systems, Man, and Cybernetics, v. SMC-14, n. 2, 1984.
2. "On Measuring the Complexity of Monitoring and Controlling Large Scale Systems," R.L. Henneman and W.B. Rouse, IEEE Transactions on Systems, Man, and Cybernetics, to appear.



ARI: Bimonthly Report      December 1, 1985 - January 31, 1986

Work during this period has continued the development of the model of human performance in CAIN. A working model now exists, although the rules and the procedure for ranking rules needs to be refined. The model is being implemented on the VAX 11/780, which is a change from the proposed idea to implement it on the AT&T 3B2. The floating point processor on the 3B2 is too slow to handle the real-time processing needs of the simulation and the model.

We plan soon to evaluate the model by comparing its performance with that of human subjects who participated in the most recent experiment. Both open-loop and action-by-action comparisons will be made.

As of January 31, 1986, approximately \$393,000 had been spent.

# CENTER FOR MAN-MACHINE SYSTEMS RESEARCH

SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING

ATLANTA, GEORGIA 30332

(404) 894-3996

May 20, 1986

Dr. J. Orasanu  
Army Research Institute  
5001 Eisenhower Avenue  
Alexandria, VA 22333

Dear Dr. Orasanu:

Enclosed are four copies of our bi-monthly report for the period of  
1 February 1986 - 31 March 1986.

Sincerely,

William B. Rouse  
Professor

Enclosures

cc: F. Cochran  
Pat Heitmuller

/eml

ARI Bi-monthly Report: 1 February 1986 - 31 March 1986

Substantial progress has been made during this period towards implementing the model of human performance in CAIN. The model contains a fairly small number of rules (about 22), which seem to describe the range of possible actions in CAIN. The way in which the model selects rules is based on a fuzzy-weighting scheme that takes into account an a priori ranking of rules, the current state of the system, and the model's contextual knowledge of the system. Thus, the model's performance changes with time; as the contextual representations become richer, those aspects of performance related to the contextual knowledge will improve.

The model's contextual knowledge is represented as a set of hierarchically connected frames. Each frame contains slots of information about relevant system properties (e.g., high loading rate, recurring failure area). As the model becomes more "expert" the structure of the frame should more closely resemble that of the real system.

An experiment to validate the model and to test the effectiveness of an on-line model-based performance aid will begin in the next several weeks. Two groups of 10 subjects will be used. The first group will participate in a total of nine sessions controlling CAIN (3 training, 6 data collection). All subjects will control a 3-level, 16 node CAIN system for each of the sessions. The data collected from this group will be used to evaluate the model. (Both open-loop and action-by-action comparisons will be made.) A performance aid based on this model will then be implemented. The second group of ten subjects will control the same systems as the first group, but with the help of the aid. Performance of the aided and the unaided groups will then be compared. Independent variables, therefore, are aid vs. no aid and session number. Dependent variables will include those used for the previous experiments, including the complexity measures.

As of 31 March 1986, approximately \$402,230 had been spent.

ON MEASURING THE COMPLEXITY OF MONITORING AND CONTROLLING

LARGE SCALE SYSTEMS

Richard L. Henneman

William B. Rouse

Center for Man-Machine Systems Research

School of Industrial and Systems Engineering

Georgia Institute of Technology

Atlanta, GA 30332

ABSTRACT

The complexity of monitoring and controlling a large scale system, such as a communication network, is considered. Relevant literature is reviewed, with emphasis on both behavioral and non-behavioral approaches to measuring complexity. A simulated large scale network is described that is used in an experiment to assess the effect of network redundancy and number of system levels on human fault diagnosis performance. Experimental data is also used to evaluate two time-varying measures of task complexity (using ANOVA and time-series analysis). The first measure is dependent upon the structure of the system; the second measure is dependent on the strategy of the person controlling the system. Results suggest that this distinction is appropriate. In addition, results emphasize the different implications that complexity can have for normal system operation and human failure diagnosis performance. Although system design characteristics such as redundancy may help to avoid the short term effects of failures, these same characteristics may have the dual effect of making the human supervisory controller's task more difficult.

## INTRODUCTION

Recent trends toward increased automation in large scale engineering systems are causing a parallel shift in the role that humans play in these systems. People are increasingly being required to interact with systems only during unforeseen events, such as when a part of the system fails. During these times, proper system functioning is dependent upon the human's decision making and problem solving skills. These human abilities can be enhanced or degraded by a parallel shift in display capabilities: not only is the computer changing the level of automation in systems, but it is fundamentally changing the nature of communication between the human and the system. These changes have the potential of producing tasks of possibly enormous complexity. In light of this potential, it is of basic importance to consider human abilities in monitoring and controlling these complex environments.

Research activity over the past several years has considered the fault diagnosis abilities of humans in a supervisory control context [1], although much of this work has been confined to the process control domain. Of additional importance is the consideration of human performance in monitoring and controlling large scale hierarchical networks, such as communication or command and control systems. These systems typically can be represented as large queueing networks, with the extent of control increasing with successive hierarchic levels. Due to the enormous size of the system, not all relevant information can be displayed to the human operator at one time; thus, multi-page computer generated displays are frequently used. Human limitations in dealing with systems of this type have not been investigated to any great extent [2].

The work reported in this paper is an effort to relate aspects of system design to the complexity of the human operator's monitoring and control task. Emphasis is placed in the following section, therefore, on identifying a variety of perspectives on complexity. A simulated large scale system (an extension of the one reported in Henneman and Rouse [2]) is then described, which is used in an experiment to evaluate two dynamic measures of task complexity that are based on the structure of the system and the strategy of the human operator.

## BACKGROUND

The purpose of this section is to review discussions and investigations of complexity that have taken place within a number of disciplines. Computer scientists, for example, are often interested in the computational complexity of a particular algorithm. Computer scientists also often measure the complexity of a piece of software. General systems scientists postulate theories about the inherent complexity of large scale systems, while theoretical biologists discuss the complexity of biological systems. Psychologists relate the complexity of symbolic or spatial patterns to human behavior. Man-machine systems engineers are interested in system complexity as it relates to human problem solving and system control. In this section, the issue of complexity is addressed from these and several other perspectives. For organizational purposes, non-behavioral perspectives are considered first, followed by behavioral complexity perspectives.

### Non-behavioral perspectives

Computational complexity. An issue that has interested computer scientists, operations researchers and others is that of the relative computational difficulty of computable functions (i.e., why is one function more difficult to compute than another?). In general, computational, combinatorial, or algorithmic complexity is defined as the length of time or amount of space (memory requirements) required to compute a certain function on a certain type of machine [3,4]. Algorithms are classified in terms of the amount of time (e.g., polynomial or exponential) and/or memory they take to be solved on a computer [5,6,7]. Examples include an analysis of a graph theory algorithm for cluster analysis [8], a consideration of the complexity of mathematical models in manipulator control systems [9], some observations regarding the complexity of matrix factorization [10], and an examination of the time required to solve problems in a system of communicating sequential processes [11].

As Rouse and Rouse [12] have noted, a relatively large amount of work has been done to analyze the complexity of automatic fault detection algorithms. Fujiwara and Kinoshita [13], for example, analyze several problems of instantaneous and sequential fault diagnosis of systems. They show that these algorithms are polynomially complete (i.e., they can be solved in polynomial time if and only if the traveling salesman problem, knapsack problem, etc., can be solved in polynomial time.) Priester and Clary [14], using results from system identification theory, develop measures of failure test complexity. Rouse and Rouse

[12] try to relate human performance to an optimal solution of a fault finding task.

Software complexity. Somewhat related to the measurement of computational complexity is the measurement of software complexity. While computational complexity estimates the time and memory requirements of implementing a particular algorithm on a computer, software complexity estimates such quantities as programming time and program length. By controlling the software complexity, production costs should reduce while overall software quality should increase [15].

Halstead [16] has proposed a theory of software science that is based on a measure which counts the number of operators and operands in a program in order to estimate program length, volume, program level, language level, programming effort, and programming time. Despite a high degree of predictive power, criticism has been leveled at the approach from a theoretical perspective [17,18]. Other approaches include a graph-theory based measure of McCabe [19], an information theory based measure [15], and a control structure/flow measure [15]. Davis [20] notes that none of these approaches are based on a satisfactory model of programmer cognitive processes, and thus, proposes and evaluates measures based on "chunks", or related program concepts that can be understood by programmers, as a single cognitive unit. Chaudhary and Sahasrabudhe [21] conclude on the basis of experimental results that complexity not only involves the control structure of a program but also the executional difficulty of the program.

Complexity of physical systems. Besides the complexity of mathematical algorithms or computer software, complexity has also been discussed in the context of a physical system. Typically these



investigations are of a general, theoretical nature, although some of the discussions are applicable to the consideration of human performance in large scale systems. In the following paragraphs, the general systems approach to understanding complexity is considered.

Weaver [22] has distinguished between problems of simplicity, disorganized complexity, and organized complexity. Problems of simplicity include the largely two variable problems considered by the physical sciences before 1900. Problems of disorganized complexity contain a very large number of variables, each of which may possess an erratic or unknown behavior. By applying techniques of probability theory or statistical mechanics, the behavior of the system as a whole may be analyzed and characterized by its average tendencies. An important range of problems lies between the extremes of simplicity and disorganized complexity. These problems may contain a relatively large number of variables; however, they also exhibit a high degree of organization. Problems of organized complexity are ones in which "a sizeable number of factors ... are interrelated into an organic whole" [22]. In general, these problems are of interest to the system scientist. Systems of all types -- biological, social, economic, ecological, or physical -- can be characterized as highly interrelated subsets of variables.

Redundancy and complexity. A recurrent theme throughout the literature is the identification of system size and degree of interconnectedness as indices or attributes of system complexity. Example domains include general systems [23,24], architectural design [25], and political systems [26].

A highly connected system is complex, however, only in the sense that it is difficult for a person to understand the causal net of relations among system components and variables. Thus, a high level of connectivity (or redundancy) should lead to increased difficulty in solving problems related to system operation (i.e., failure detection, resource management, etc.). Waller [27], for example, proposes that large, highly connected systems are complex and difficult for humans to understand because of inherent human information processing limitations.

With respect to normal system control, however, the concept of redundancy has quite different implications. Mackinnon and Wearing [28] investigated a complex decision making environment in which the number of elements in the system, the degree and pattern of interconnections in the system, and the presence/lack of uncertainties in the system were varied. The results indicated that the complex (or highly interconnected) systems did not always lead to poorer levels of performance. In these cases, therefore, a high level of redundancy led to improved system performance. The authors claim that this effect is due to the insensitivity of highly redundant systems to faults and mistakes made by subjects.

Thus, at least two different interpretations of the relationship between redundancy and system complexity exist. The first interpretation, generally espoused by social scientists and general systems theorists, is related to the difficulty of understanding the system. When a failure occurs it may be difficult to locate its cause due to the presence of multiple paths through the system. On the other hand, when a system is highly redundant, its ability to carry on normal operation is greatly increased — the redundancy serves to stabilize the

network. This interpretation is largely held by biologists and engineers. Thus, the level of interconnectedness in a system affects the level of two types of complexity: problem solving complexity and system control complexity.

These two interpretations are consistent with standard results from reliability theory [29,30,31]. As the number of alternate paths (or components) increases in a system, the reliability increases, as expressed by the mean time between failures. However, data has shown that as the redundancy (and hence, the reliability) increases, the maintainability of the system decreases, i.e., the mean time to repair increases [30]. Thus, more complex (or redundant) systems lead to longer repair times. The availability of the system (or the probability that the system is operating satisfactorily at any point in time) is shown by von Alven [31] to be a function of both reliability and maintainability; thus, it too is a function of system redundancy.

Subjective nature of complexity. A final theme within the complexity literature is that of the relative or subjective nature of complexity. Ashby [32] illustrates this concept by considering a sheep's brain. While the internal mechanisms of the brain are very complex to a neurophysiologist, a butcher only has to distinguish a sheep's brain from about 30 other cuts of meat (or about 5 bits). Several other authors also equate complexity with descriptions of objects, rather than with intrinsic properties of objects [33,34,35]. This perspective leads quite naturally to the discussion of behavioral complexity which is pursued below.

Summary. The following conclusions can be made on the basis of the review so far:

1. Complexity is related to the size of the system as well as the level of redundancy (or connectivity) among components.
2. The effects of redundancy on complexity differ depending upon one's perspective. A highly redundant system may lead to better overall performance; however, it may also lead to increased human problem solving difficulty.
3. Complexity can only be measured relative to a person's understanding of the system.

#### Behavioral complexity

The preceding discussion has made only oblique reference to human abilities in perceiving information about the system or in solving problems within the environment created by the system. From a psychological perspective, the relationship between complexity and human performance is of fundamental importance. This relationship is explored in the following sections. Perceptual complexity is considered first, followed by problem solving complexity.

Perceptual complexity. Rouse and Rouse [12] describe studies of perceptual complexity as dealing with "... the human's ability to recognize, rotate, reverse, etc. displayed patterns as a function of various attributes of the pattern, including number of line segments, symmetry, etc." This form of complexity has typically been investigated via some simple experimental scenarios. Greenberg and Krueger [36], for example, use a letter searching task to examine the relationship between task difficulty (in terms of letter orientation and redundancy) and speed of search. Other studies examine such aspects of complexity as color

[37], stress and its relation to a visual discrimination task [38], and relations between visual complexity and verbal associative value [39].

Hochberg and Brooks [40] derive a complexity measure based on the number of angles, number of lines, and the variety of angles contained within a drawing. Vitz and Todd [41] also propose a complexity metric of non-representational shapes based on a sampling of elements in the drawing. Butler [42] extends this work by using a complexity measure based on information load and the number of lines in the drawing. Attneave [43] develops a complexity measure based on the physical characteristics of shapes. Kimchi and Palmer [44] relate the number of elements in a drawing and its size to subjects' similarity judgements and their verbal descriptions. Finally, Simon [45] reviews several different approaches to relating the perceptual complexity of patterned sequences of symbols to human behavior. Simon concludes that all of the theories share a common central core: subjects perform the tasks by inducing pattern descriptions from the sequences. These descriptions all involve the same rules between symbols, iteration of subpatterns, and a hierarchic phrase structure.

Relative to the role perception plays in the complexity of fault diagnosis tasks, Rouse and Rouse [12], in their study of complexity measures of fault diagnosis tasks, use the number of displayed components as a measure of perceptual complexity. Results indicate that this measure is not a good predictor of fault diagnosis performance. Since the number of components displayed on the screen is a function of the equipment's inherent complexity, not peculiarities of the display, the authors advise that a systematic variation of display characteristics might indicate that fault diagnosis tasks can be perceptually complex.

In light of the success of other predictors which are more related to problem solving complexity that are discussed in the next section, the authors suggest that problem solving measures are more relevant to fault diagnosis tasks.

Brooke and Duncan [46] extend the work of Rouse and Rouse [12] to examine explicitly the effect of display formatting on measures of the fault diagnosis process. Results indicate that changing some of the perceptual characteristics of the display improves the speed and diagnostic efficiency with which faults are located.

Problem solving complexity. A second form of behavioral complexity, which has received less attention than perceptual complexity, is problem solving complexity. This type of complexity measure assesses various problem attributes and attempts to relate them to human reasoning abilities and problem solving skills. Experimental assessments of problem solving complexity typically use syntactic or arithmetic problem solving tasks. Glover et al. [47], using a written learning task, finds that more difficult tasks result in higher levels of recall. McDaniel [48] reports that syntactically complicated sentences result in greater recall of sentence structure than do simple sentences. Ashcraft and Stazyk [49], using mental arithmetic tasks, discover that reaction time increases with increasing problem complexity. Loftus and Suppes [50] find that problem solving difficulty of arithmetic word problems is related to problem attributes like surface structure, number of words, and the number of different operations required to obtain a solution. Morgan and Alluisi [51], using a code transformation task, find that problem complexity has a greater effect on performance after practice than the early trials.

Kieras and Polson [52] discuss "user complexity," which is the complexity of a device or system from the point of view of the user. The authors propose that user complexity depends on the "amount, content, and structure of knowledge required to operate a device." In addition, the complexity for a novice increases as a function of the difficulty of acquiring that knowledge. Knowledge is composed of two components, task knowledge and device knowledge. Complexity, therefore, is dependent not only on device or task characteristics, but also on the knowledge of the user. In order to measure complexity, the authors suggest the following indices: number of productions (rules) to be learned, number of productions fired, number of keystrokes, number of items in working memory, etc. The authors propose that these measures of user complexity can be determined by using a computer simulation to implement a user model.

With respect to measures of problem solving complexity in man-machine systems, the most pertinent work is that of Rouse and Rouse [12]. Besides their measures of number of components and optimal solution which have already been discussed, Rouse and Rouse also propose two measures of problem solving complexity: the number of relevant relationships (i.e., number of possible causes of a set of symptoms) and an information theoretic approach. These two measures are highly correlated with human performance in the fault diagnosis tasks (as measured by time to solution). The authors suggest that the success of these measures can be largely explained by the fact that they reflect the human's understanding of the problem and his resulting solution strategy.

Wohl [53,54,55] examines the relation between the structure of electronic equipment and human fault diagnosis performance. He derives a

measure of complexity based on system connectivity which is shown to predict repair times very well. Wohl relates this measure to human cognitive limitations. He suggests that if some upper bound of complexity is reached (namely, human short term memory limits), some fraction of equipment failures will be non-diagnosable. Existing equipment does not exceed these human cognitive limits since designers as well as diagnosticians possess the same limits. However, these results have rather important implications for computer-aided design, which could allow the creation of overly connected parts. It should be noted that although this measure is related to the Rouse and Rouse measures of complexity, it differs because it reflects mostly characteristics of the system rather than characteristics of the human.

Summary. The following conclusions can be made on the basis of the review of the behavioral complexity literature.

1. Measures of problem solving complexity appear to be most relevant to the task of failure diagnosis, although perceptual complexity may play some part in affecting task difficulty.
2. Complexity is caused not only by the attributes of the problem solving environment, but also by the human's understanding or perception of those attributes.
3. Little work has assessed the complexity of problem solving in large-scale man-machine systems.

#### Implications of complexity

It is reasonable to assume that complexity should manifest itself in some measurable way; i.e., a complex system should result in longer times to failure diagnosis, longer reaction times, etc. In order to validate a



complexity measure, it is important to identify correctly and to justify an appropriate dependent measure.

A survey was made of 19 behaviorally oriented studies of complexity reviewed in this section. The most popular dependent measure (eight) was reaction time or time to problem solution. Other dependent measures were solution success, recall of sentence structure, memory of forms, and dimensionality judgements of figures. Three studies used number of errors as the dependent measure. Few of the studies, however, (other than Rouse and Rouse [12]) offer any rationale for their choice of a dependent measure.

### Conclusions

The preceding sections have considered definitions, measures, and implications of complexity within a variety of domains. On the basis of this review, it is instructive to make some generalizations.

Most studies of complexity performed by systems scientists are on a general level. Although much work has gone into defining and measuring system complexity, little has been done to assess the implications of complexity. Furthermore, assuming that humans must play an important role in many large scale systems (e.g., failure diagnosis and network management), little research has investigated the relationship between large scale system complexity and human performance. Due to the strong theoretical flavor of this approach, it is often difficult to see its application to real world systems.

On the other hand, studies of complexity performed by behavioral scientists are on a very applied level. Although the approach often lacks the theoretical rigor of the systems approach, complexity is always

related to some aspect of human performance. Unfortunately, differences between tasks and complexity measures make it difficult to generalize results across contexts. Moreover, the small, well-defined nature of the tasks seems to have little relation to human performance in large scale systems.

The remainder of this paper is devoted to consideration of human performance in monitoring and controlling large scale systems. Thus, the research attempts to integrate a number of the issues raised in this section concerning the nature of complexity. Complexity is viewed as being a result of both the structure of the system and the human operator's understanding of the system. Complexity is also considered in terms of its relation to both system performance and human performance. In particular, the relationship between such structural variables as redundancy and number of levels and performance is investigated. In summary, the goal of this work is to "bridge the gap" between systems science and behavioral science and, in the process, gain practical insights into appropriate roles for humans in the increasingly complex systems that technology is producing.

#### TASK DESCRIPTION

A previous experiment [2,56] considered human performance in the monitoring and control of an essentially context-free representation of a large scale system. Subjects monitored and controlled a computer simulated large scale system called MABEL (Monitoring, Accessing, Browsing, and Evaluating Limits), trying to optimize such system parameters as number of customers served and customer processing time while trying to diagnose system failures. As noted in the Background

section, of interest is the assessment of measures of task complexity; i.e., what features of the physical system, the human-system interface, or the human's understanding of the system make the monitoring and control task difficult? A major goal of this paper is to consider the nature of complexity in a large scale system.

The remainder of this section describes a contextually augmented version of MABEL that contains substantially higher fidelity than the earlier simulation. An experiment is then described, from which data are analyzed using the same set of performance measures as were applied to the experiment reported in Henneman and Rouse [2]. Data are then analyzed from the standpoint of assessing task complexity.

#### Overview of CAIN

Certain features of MABEL were substantially changed to develop CAIN (Contextually Augmented Integrated Network); however, the underlying structure of CAIN is identical to that of MABEL. This section summarizes the similarities between the context-free MABEL and the contextually-augmented CAIN. The summary is only a very broad overview; the reader is referred to Henneman and Rouse [2] or Henneman [56] for much more detail concerning the underlying structure of the two simulations.

CAIN is programmed in Pascal on a VAX 11/780 computer and operates in real time. It is structured as a large hierarchical network that can range in size from hundreds to thousands of nodes. Customers travel through the system from a randomly selected source node to a random destination. Subjects monitor this system activity via a CRT display. When they detect a problem in the system (possibly due to a failure), subjects issue an appropriate command through a keyboard to correct and

compensate for the abnormal situation. The overall objectives of the operator are:

- 1) to maximize the number of customers served, and
- 2) to minimize the time it takes for customers to travel between source and destination nodes.

Because there are so many nodes in the network, it is not possible to display information about all nodes at one time. Thus, nodes are grouped into relatively small networks called clusters. Human operators are restricted to viewing only one cluster at a time on the CAIN display. Clusters are grouped into hierarchic levels.

#### Effects of Node Failures

Under normal circumstances, CAIN operates automatically without interference from the human operator. Since the system cannot automatically diagnose and repair failures, the human must monitor the system looking for evidence of failed components. Node failures can occur in two ways. The first is a randomly occurring failure mode caused by malfunctioning equipment. The second type, capacity failure, can be caused by the randomly occurring failures. Each node has a maximum number of customers that it can store at one time. If this limit is exceeded, the node fails. Thus, if a node fails randomly and a customer needs to visit that node, it will be retained at its previous node. This retention will cause the previous node to stop processing customers, which can lead to a capacity failure. In this way, if the operator does not quickly locate failures, the problems will propagate through the system.

### Addition of Context

Although the physical hierarchical structure of MABEL was preserved, the addition of contextual information to CAIN required changing some interface characteristics. In the MABEL scenario, for example, all nodes on a display page are identified by a number on the CRT display. Each displayed node in a cluster, therefore, is physically identical to nodes in other clusters. The MABEL interface has a generic quality in that all subsystems are visually similar; no contextual cues exist. On the other hand, nodes in CAIN are identified via specific geographic locations. Thus, a node in MABEL with the label "9" might be labelled "Chicago" in CAIN. A typical CAIN display is shown in Figure 1.

Simply introducing geographic names as node labels is not enough, however, to alter subject task performance. A small experiment ( $n=3$ ) replicated the first MABEL experiment [2,56], with the exception that nodes were given geographic names. Subjects still referred to nodes by number only; contextual labels were present but not needed to perform the task. No significant difference was found in terms of performance between subjects using the two task scenarios. This result suggests that the addition of context must be such that it provides associative links (i.e., memory aids) or cues (i.e., clues to the location of problems within the system) through which subject performance is enhanced or task difficulty is decreased.

Associative Links. The formation of associative links in CAIN is facilitated by the way in which a subject identifies a node. In CAIN, nodes are referred to by geographic labels only, never by number. Subjects may input the shortest string of characters that uniquely

identifies the node from all other nodes in the system. Thus, "Denver" may be abbreviated "den". Most nodes can be identified with a three or four character substring of the complete name. In addition, the number of elements on a display page is kept constant at 16 so that the contextual information is invariant.

To illustrate the effect this change has on the subject's task, consider the command that displays a lower level cluster. In MABEL, the subject inputs the command "d2", which displays the cluster beneath Node 2. In CAIN, on the other hand, the subject types "dSanf", which displays the cities beneath San Francisco (e.g., Berkeley, San Jose). Thus, subjects can form associations or links between system parts due to the existence of contextual information.

Subjects can use these learned associative links to maneuver through the CAIN display hierarchy. In MABEL, movement between display pages is constrained to the cluster of nodes immediately above or below the current display. Thus, it is not possible to jump laterally across the network. In CAIN, however, it is possible to move from one part of the system to any other part. For example, if a subject recalls that the cluster associated with Bangor, ME was previously experiencing problems, it is relatively easy to call up that cluster display. This is done by using a "find" command ("f"). In addition, subjects can return immediately to the highest level in the system by inputting the "a" command. (A complete list of commands available for use in CAIN may be found in Table 1. This command list is categorized by function: access, monitor, diagnose, or control.)

Cues. The formation of cues in CAIN is provided by the introduction of context-dependent events. These events are of one of two types:

recurring failures and non-uniform loading. Although equipment in nodes fails randomly, some equipment experiences a higher probability of failure. For example, a thunderstorm in Little Rock, AR may make equipment in that city susceptible to lightning damage. Similarly, given that incidents of vandalism are more likely to occur in Newark, NJ than in Council Bluffs, IA, there is a greater chance of equipment damage in Newark. Therefore, equipment in certain cities exhibits a greater tendency to fail than in other cities. Subjects are informed of these locations via warning alarms that appear on the bottom of the display. Subjects can directly monitor activities within these trouble spots via a special "watch" ("w") command. Subjects acknowledge the alarms by inputting an "erase" command ("e"). Subjects add and delete trouble areas from the watch list by using "+" and "-" commands.

Besides recurring failures, another type of context-dependent event present in CAIN is non-uniform loading. At different times, certain sections of the system may be prone to experience heavy loading. For example, certain times of day are busier in one part of the country than in others. Similarly, a major political or sports event in one section of the country may increase the number of messages sent. As with the recurring failures, subjects are told the location of these increased loads via a message at the bottom of the screen. Subjects can reduce the number of customers admitted to the overloaded subsystem by means of the "load" ("l") command.

In summary, despite the structural isomorphism of the two simulations, CAIN represents a significant departure from the context-free scenario of MABEL. Through the addition of contextual detail and the addition of events that are dependent upon this contextual

information, the simulation fidelity has been increased significantly.

#### MEASURES OF COMPLEXITY

The Background section considered complexity from non-behavioral and behavioral perspectives. When assessing the complexity of an operator's task in monitoring and controlling a large scale system, both approaches should be taken into account. In this paper, therefore, the complexity of a large scale system is described in terms of: 1) the physical structure of the system and, 2) the operator's understanding of the system as reflected by his strategy. From this perspective, a system that is complex or difficult to control for one operator may be relatively easy to control for another operator. Similarly, the complexity of a system may vary with time for any particular operator. Some systems, however, may be complex regardless of any particular control strategy due to their inherent structural complexity. The following paragraphs propose two measures of complexity that incorporate these ideas. Structural complexity is considered first, followed by strategic complexity.

##### Structural Complexity

A one-to-one relationship exists between the hypothetical physical structure of CAIN and the actual structure of the display page hierarchy. Since the main control task in CAIN is to locate failures, a measure of structural complexity should assess the difficulty of finding failures given the physical arrangement of the system. A major constraint placed on an operator's ability to locate failures is the hierarchical display structure; thus, it seems reasonable to assert that structural complexity



can be estimated by calculating the total number of display pages the operator must view in order to repair all system failures. Assuming that the operator knows the location of all failures, this measure represents the minimum number of pages necessary to locate all system failures. Thus, the structural complexity measure represents optimal performance given the constraints of the structure or arrangement of the system components. Operator performance affects this measure only in that any particular operator may have more or fewer failures depending upon his fault finding ability.

To illustrate how this measure is calculated, consider the system in Figure 2. This hypothetical system contains four nodes per display page and has three levels. Each group of four rectangles represents a cluster of nodes (i.e., one display page). For clarity, only those clusters of nodes that enter into the complexity calculation are shown. The darkened rectangles represent nodes that have failed. In this example, three failures exist within the system: two on the second level and one on the third level.

The structural complexity measure is determined by counting the number of display pages that must be viewed in order to find all failures. The counting method assumes a strategy based on tracing higher level symptoms to their causes in the lower levels. (Context-specific cues might, of course, allow operators to locate failures in fewer pages.) Thus, the counting method assumes that after locating all failures along one subsystem branch, the subject returns to the highest system level to search the next branch (a depth-first strategy). Figure 2 is self-explanatory; to repair all three failures in the system, an operator must view at least six display pages. The final return to the

top system level is not counted into the measure because it would simply add 1 to all estimates.

### Strategic Complexity

The strategic complexity measure explicitly considers operator performance. When an operator is deciding which path through the system is most likely to lead to finding a failure, he makes a tradeoff between his uncertainty concerning the state (i.e., queue lengths) of a subsystem display page and his expectations of finding a failure in that subsystem. High uncertainty about a subsystem may be acceptable, for example, if a relatively low probability exists of finding a failure on that display page. On the other hand, high subsystem uncertainty may be unacceptable if a very high probability exists of finding a failure. These observations suggest that an appropriate measure of strategic complexity that reflects the trade-off between state uncertainty and probability of failure is the multiplication of these two metrics.

State uncertainty ( $U$ ) is defined as the real time elapsed since a particular display page was last tested for failures. Probability of failure is defined as the probability that a failure exists within a cluster given the state of the display ( $p[F|X]$ ). For example, when a subject views a particular display page, features of that display provide information about the existence of failures in other subsystems (e.g., a large queue size suggests a lower level failure.) Experimental data files were replayed in order to estimate these probabilities empirically. These probabilities were determined by dividing the frequency with which a display state reflected a failure by the frequency with which a particular display state was viewed by an operator. Sets of

probabilities were calculated for different system configurations (2 vs. 3 levels and high vs. low redundancy), and different loading rates (e.g., a system with a low loading rate has fewer customers in service, and hence, lower queue sizes will reflect failures).

The measure of strategic complexity multiplies these two measures (state uncertainty and probability of failure given the system state) and sums the product across all clusters in the system:

$$\text{Strategic Complexity} = \sum_i U(i) \times p[F|X(i)]$$

where  $U(i)$  = time since last accessing display page  $i$

$X(i)$  = State of page  $i$  reflected by display one level higher

$p[F|X(i)]$  = probability of failure given state  $i$

and  $F$  denotes "failure"

When a subject descended to a lower level, the  $p[F|X(i)]$  remained fixed for the previous level. When a subject returned to the higher level, the  $p[F|X(i)]$  values associated with the just-visited lower level cluster were set to zero. Thus, when an operator descended to a lower level subsystem and tested for failures, the strategic complexity measure was simultaneously increased by the "new" uncertainty present in the other lower-level subsystems and decreased by the certainty now associated with the current level.

To illustrate how the strategic complexity measure is determined, consider the display in Figure 3. This system contains four nodes per display page and has two levels. The operator is viewing the highest

level page in the display hierarchy and is monitoring activity in the next level of the system. The operator can gather information about activity in the second level of the system from two sources in this example: the cluster display and the data displayed via the monitor command. The monitor command lists the number of customers in the clusters one level below; the cluster display shows the number of customers waiting at all nodes in the current cluster.

Each of these pieces of information reflects the probability that a failure has occurred in a lower level cluster. These probabilities (which are plausible, but hypothetical) are listed in Table 2. For example, the queue size of 15 in Denver reflects a relatively high probability (0.75) that a failure exists in Level Two. Similarly, the monitor command reports that eight customers are currently in the cluster beneath Denver; these eight customers reflect a 0.60 probability that a failure exists. The operator has not tested the cluster beneath Denver for failures for  $U(\text{Denver}) = 20.12$  seconds. Using the information that reflects the highest probability of failure (i.e., from the cluster display) results in the following measure of strategic complexity for the Denver region:

$$\begin{aligned} U(\text{Denver}) \times p[F|x(\text{Denver})] &= 20.12 \times 0.75 \\ &= 15.09s \end{aligned}$$

This procedure is then repeated for the other clusters in the network and the measures are added together. In this way, the total strategic complexity is determined to be 15.61.

In this example, it should be noted that Denver makes a very large contribution to the strategic complexity measure as a result of two factors: first, the operator has a high degree of uncertainty concerning the Denver subsystem in that he has not tested that cluster for failures in 20.12s. Second, the display reflects a very high probability (0.75) that a failure exists in the Denver subsystem. The combination of these two factors leads to a very high measure of strategic complexity for the Denver subsystem. On the other hand, the other subsystems have either a low uncertainty measure or a low probability of failure. Thus, their contribution to strategic complexity is small.

Finally, it is instructive to consider the extent to which an operator may "optimally" reduce strategic complexity. Since the measure is based on time, it will continually increase unless either the operator performs some action or the state of the system shifts. At any instant in time, therefore, it is possible for an operator to reduce strategic complexity optimally by viewing the display page that reduces the measure by the largest amount (i.e., the cluster with the largest  $U \times P[F|X]$  value). In the long run, however, the measure may only be optimally reduced given the operator's performance constraints (i.e., psychomotor reaction and movement times). In other words, since the measure will in general keep increasing with time, optimal performance will always be limited by how long it takes the operator to physically select the next display page.

#### Dependent Measure of Complexity

The literature review also suggested that an appropriate dependent measure of complexity is the time until failure diagnosis. In the

context of CAIN, this measure is the average time until the subject issues a repair command for a failed node. Since the two independent complexity measures vary with time, it was necessary to use a dependent measure that also changes with time. Average time, therefore, includes the diagnosis time for the current repair plus diagnosis times for the four previous repairs.

#### Summary of Complexity Measures

To summarize, the structural measure reflects an inherent characteristic of the network, namely the number of display pages necessary to find all of the failures in the system. The strategic measure, on the other hand, reflects temporal aspects of subjects' strategies, i.e., subjects' paths through the network. From this perspective, the strategic measure reflects the complexity resulting from a particular strategy.

Although the two complexity measures proposed here may have some general applicability (in particular, the measure of strategic complexity is appealing due to its temporal nature), it is not the intent of this paper to suggest or prove that these measures are true indices of task complexity. The goal instead is to show in a pragmatic sense that these two dimensions represent a useful distinction relative to task complexity. These measures represent a convenient means to demonstrate this distinction.

## METHOD

### Motivation

The main goal of this experiment was to investigate the nature of complexity in a large scale human-machine system. As emphasized in the preceding section, the general assumption is made that task complexity can only be measured relative to an individual's understanding of the system and his expertise in dealing with problems in that system. Thus, complexity is considered to be dynamic, varying across time and among subjects. Accordingly, as discussed below, subjects were required to perform the task (CAIN) over a relatively long period of time.

### Subjects

Eight junior and senior engineering majors at Georgia Tech served as subjects in this experiment. Due to the nature of the task, potential subjects were screened via a typing test (minimum ability level was 25 words/minute). Subjects were paid a total of \$65: \$5.00 for each training session (3) and each experimental session (10).

### Training

Subjects were trained via a combination of written instructions and hands-on experience with CAIN. Subjects initially were given two sets of written instruction on consecutive days explaining the system, the goals of their task, and methods for achieving these goals. Self-test questions were contained within the text to insure mastery of the material. The experimenter reviewed this material with subjects at the beginning of each training session. In addition, subjects were given one-page summaries detailing the structure of the system, available commands (Table 1), and operation of the system.

Subjects completed the first two training sessions by controlling a two-level CAIN system. The third training session was spent controlling a three-level CAIN system. These sessions were performed using a version of CAIN that allowed subjects to start and stop the program execution. Thus, subjects could investigate normal and abnormal system functions without being overwhelmed by the progressive effects of failures. The experimenter was present during all training sessions to answer questions.

### Experimental Design

Henneman and Rouse [2] reported that cluster size (number of nodes per display page) in MABEL had a particularly strong effect on task performance. Results suggested that small clusters degraded performance because fewer connections existed between nodes; less redundancy caused failures to propagate more quickly. Another result from Henneman and Rouse [2] showed the very strong effect of number of hierarchical system levels on human performance. Increasing the number of levels from two to three degraded performance. Thus, two independent variables selected for further analysis were the degree of redundancy (or connectivity) and the number of levels in the system. (Cluster size was kept constant at 16 as mentioned previously in order to emphasize the non-varying features of the contextual display.) Redundancy or connectivity was defined as the number of connections emanating from each node. Redundancy varied between low (6 connections/node) and high (13 connections/node) and number of levels varied between two and three.

Of interest in this experiment was the way in which complexity changes as subjects gain expertise. Thus, the order of presentation of experimental conditions was not randomized. All subjects saw the same



experimental conditions in the same order. A final independent variable, therefore, was the order of presentation of experimental conditions.

In summary, the ten experimental sessions (S1 - S10) were performed in the following order (with the intent of increasing experimental difficulty): S1,S2: 2 levels, high redundancy; S3,S4,S5: 3 levels, high redundancy; S6,S7: 2 levels, low redundancy; S8,S9,S10: 3 levels, low redundancy. Each experimental session was performed on consecutive days and lasted about 45 minutes.

## RESULTS

### Summary of Approach

Data from this experiment were first analyzed using the same performance measures as the experiment reported in Henneman and Rouse [2]. Overall results from the analysis of variance supported those of the earlier experiment. In light of this similarity, these general results are only briefly summarized below. Considerably more detail may be found in Henneman [56].

Measures of fault diagnosis performance were affected as expected by the independent variables. Increasing the number of system levels from two to three corresponded to a higher average time to failure diagnosis. This result was largely because failures take longer to propagate upwards in the 3-level systems. In addition, failure-related symptoms take longer to emerge in highly interconnected networks; thus, the high redundancy systems resulted in longer average times to diagnosis.

The fraction of failures repaired by subjects was also significantly affected by increasing the number of levels: as the number of levels increased from two to three, the fraction of failures found decreased

from 0.95 to 0.69. As in Henneman and Rouse [2], subjects could not cope with the very large search space in the three level systems.

Data were also analyzed with the purpose of investigating relationships between the complexity measures, the CAIN environment and operator performance. This investigation was accomplished in two ways. First, an analysis was undertaken of average or global measures of complexity (i.e., the complexity time series averaged over each experimental run). The effect of the experimental independent variables (number of levels and degree of interconnectivity between nodes) on the average complexity measures was determined by using analysis of variance. The relationship between the average complexity measures and measures of subject fault diagnosis performance was then assessed by using correlation analysis. As is discussed below, this analysis of average complexity values provided explanations for differences that exist between different system configurations.

The second way in which complexity was investigated involved using a fine-grained approach, namely, time series analysis. Time series analysis was selected due to the intrinsic time-varying nature of the independent and dependent complexity measures. As will be seen, this analysis provided insight into the way in which complexity evolves and affects different phases of the failure diagnosis process.

Due to the amount of time necessary to perform these analyses, the results are limited to Sessions 2,5,7, and 10. Data for the analyses were generated by replaying subject data files. Following every three seconds (corresponding to the rate of display update), both complexity measures and the average time until failure diagnosis were calculated. Average values for all measures were calculated from these time series.

### Analysis of Global Complexity Measures

Analysis of Variance. The results of two ANOVAs (using average structural and strategic complexity measures as dependent measures and number of levels and degree of redundancy as independent measures) are qualitatively summarized in Figure 4. (Henneman [56] reports the results more fully.) Structural complexity, as measured here, may be decreased in two ways: 1) decreasing the number of system levels and 2) decreasing the number of system failures.

The first way (decreasing number of system levels), enables subjects to access fewer display pages in order to diagnose failures in the lowest system level. The second way (decreasing number of system failures) is facilitated by increasing the network redundancy (i.e., increasing the number of connections between nodes). As network redundancy increases, the average number of node capacity failures decreases, which has the effect of decreasing the structural complexity measure.

Strategic complexity, as measured here, may be decreased in three ways: 1) utilizing an effective strategy in terms of responding to symptoms, 2) decreasing redundancy, and 3) decreasing number of levels (which causes symptoms to emerge more rapidly). Subjects tended to trace failures to the lowest system level only when a symptom (i.e., visual cue) appeared on the display, even if they had not viewed a particular region in a large period of time. Consequently, when symptoms emerged slowly (as in the high redundancy/three level conditions), high uncertainty resulted. This uncertainty helped to create moderate to high strategic complexity. On the other hand, symptoms emerged more rapidly in the low redundancy/two level conditions. Since operators tended to

respond primarily to visual symptoms, low redundancy led to low values of strategic complexity.

This dependence on visual cues has implications for the design of task performance aids. Aids should help people to overcome their inability or reluctance to reduce system uncertainty despite the absence of failure symptoms. Alternatively, cues or symptoms could be enhanced so that operators naturally pursue leads sooner.

In summary, increasing redundancy (or number of connections between nodes) led to less structural complexity but more strategic complexity. This result reflects findings from the literature: more redundant systems (corresponding to less structural complexity) enhance the proper operation of the system by reducing the impact of failed components. On the other hand, more redundancy leads to increased strategic complexity (the complexity of failure diagnosis) due to the slower emergence of failure symptoms.

In addition, increasing the number of system levels increased both types of complexity. Again, although multiple system levels might be desirable in that they allow supervision of larger networks and protect upper levels from the effects of failures, they have the undesirable side effect of masking symptoms from operators, thereby increasing the complexity of failure diagnosis. Multiple displays could possibly be used to reduce this complexity.

Correlation Analysis. Pearson product-moment correlation coefficients were calculated between the two average complexity measures and the two dependent measures (fraction failures diagnosed and average time to failure diagnosis). Results are qualitatively summarized in this section; again, Henneman [56] contains more detail. Since significant

interaction effects due to the experimental conditions were found, the analysis was limited to comparisons among correlation coefficients within each experimental condition. Major differences among coefficients were only noted when comparing across the number of levels variable. These results are qualitatively tabulated in Figure 5.

Considering structural complexity first, the measures for both two and three level systems correlate negatively with the fraction of failures found (correlations range between  $-.31$  and  $-.88$ ). Thus, when many system failures are present on the average (as suggested by a high structural measure), a smaller fraction of failures are found. With respect to the structural measure and average time to failure diagnosis, no significant correlation exists for the two level systems, while high negative correlations ( $-.63$  and  $-.70$ ) exist for the three-level systems. In other words, high structural complexity in the three-level systems led to shorter failure diagnosis times. This result, being somewhat counter-intuitive, is caused by the following chain of events. High structural complexity is caused by a large number of failures, which are caused, in turn, by a high number of capacity failures. Most capacity failures are located in the upper system levels where failure diagnosis times are relatively short.

The correlations associated with the strategic complexity measure tend to be smaller. Correlations between average strategic complexity and percent failures diagnosed are negative for two level systems ( $-.34$  and  $-.49$ ) and positive for three level systems ( $.25$  and  $.49$ ). In the two-level conditions, therefore, high strategic complexity led to fewer diagnosed failures, although in the three level systems, high strategic complexity led to more diagnosed failures. Results for the two level

systems are as expected. High values of strategic complexity resulted from high uncertainty and high conditional failure probabilities. Apparently subjects who used strategies that tolerated these high values were not looking at or using the display cues to find failures; thus, they found few system failures.

Results for the three level systems are less intuitive. Subjects who found many failures in the three level systems had to spend time accessing third level subsystems. Because they spent more time in the third level, these better subjects had to tolerate greater uncertainty about the rest of the system. This increased uncertainty had the effect of increasing the strategic complexity measure.

Summary. In summary, the results presented in this section provide insight to the overall characteristics of the two complexity measures and their relationship to subject fault diagnosis performance. The measures are sensitive to variations among the system characteristics of number of levels and degree of redundancy. In general, the more complex systems have three rather than two levels. The effect of redundancy on complexity depends on the type of complexity: low redundancy networks result in more structural complexity; high redundancy networks result in more strategic complexity.

An important conceptual and methodological issue raised by these results concerns the multidimensional nature of complexity. In particular, the relationship between the independent and dependent measures of complexity is of interest. When many failures exist in a system, the general tendency is for the complexity measures to increase. At the same time, however, the average time to failure diagnosis

decreases. Thus, even though complexity may be large, failure diagnosis time may be small.

This observation emphasizes the distinction mentioned previously between proper system functioning and the complexity of failure diagnosis. In a localized sense, control in a complex system is simple: no matter what the operator does, he will find a problem. This is reflected by short diagnosis times. In a global sense, however, control in a complex system is complex: so many problems exist in the system that proper operation is endangered. This is reflected by a low fraction of failures found. The operator, dealing with only a small part of the system at one time, may be oblivious to the scope of problems in the network. Another important issue is, therefore, the impact of a richly interconnected, multiple-level system (that supports proper system functioning) on the complexity of human monitoring and control (that will degrade failure diagnosis performance).

#### Analysis of Fine-Grained Complexity Measures

Time Series Analysis. Time series analysis was used to identify, estimate, and diagnostically check transfer functions that relate the two input complexity measures to the average time to failure diagnosis. The general approach is discussed by Box and Jenkins [57]. Each transfer function model predicts the current average time to failure diagnosis through a linear combination of the complexity measures at various time lags. The essence of the modeling process is to determine the time lags to include in the model and the weight or relative contribution of each time lagged variable to the predicted value. Montgomery and Weatherby [58] provide a good tutorial on multiple input transfer function models.

Transfer functions for each subject were developed for Sessions 2,5,7, and 10. (Due to space considerations, these functions are not shown here; the reader is referred to Henneman [56] for more detail.) Overall, the approach was successful. The equations remove all structure from the autocorrelation function of the model residuals. Furthermore, a comparison of the sum of squares of the original dependent time series (i.e., average time to failure diagnosis) to the sum of squares of the residuals shows that the transfer functions explain 82% to 97% of the variance within the original data. Nevertheless, wide differences in the lag and coefficient values in the models exist among both subjects and systems.

The remainder of this section is devoted to the development of a consistent explanation for these differences. The goal is not to account fully for each parameter, lag value, and coefficient in the transfer functions. Instead, the goal is to suggest a plausible explanation for the transfer function characteristics and to suggest reasons for deviations from this explanation.

Explanation for Transfer Functions. The initial step was to identify characteristics of the task, the system, or the human that could explain differences among transfer functions (e.g., long lags and inconsistency of numerical signs). For example, several different events are associated with the life cycle of each system failure: failure occurrence, symptom emergence, and failure diagnosis. Failure occurrence is defined as the time when a part of the system fails; symptom emergence is defined as the time a failure first affects any node that appears on the subject's video display; failure diagnosis is defined as the time a subject issues a repair command for a failed component. The timing of



these events undoubtedly has some effect on the length of time needed to find the failure. Moreover, the system complexity at these event times might also affect failure diagnosis time.

Besides the possibility that different events associated with the failure life-cycle impact diagnosis time, it is also reasonable that different types of diagnosis might affect failure diagnosis time. The diagnosis of any particular failure may be classified into one of three types: topographic, symptomatic, or serendipitous. Subjects identifying failures using a topographic strategy trace failure symptoms from higher system levels to their causes in lower levels. Subjects identifying failures using a symptomatic strategy make a direct mapping from their knowledge of the system structure to the failed component. A symptomatic diagnosis relies, therefore, on the subject's contextual knowledge of the system. For example, when subjects make a jump from one cluster to another cluster in the same level to repair a failure, their action suggests that their context-specific knowledge of the system is providing guidance to system trouble areas. Finally, subjects may also identify failures accidentally or serendipitously. In this diagnosis mode, subjects locate failures while browsing through the system or while tracing the cause of a different failure.

In summary, it is possible that several different types of failure-related event (e.g., failure occurrence and symptom emergence) and several different modes of failure diagnosis (e.g., symptomatic, topographic, and serendipitous) can affect the time to failure diagnosis within a system. In addition, due to the aforementioned aggregation of five failure diagnosis times for the dependent complexity measure, it is possible for many lags (possibly quite long) to enter into the transfer

functions. From the perspective offered in the preceding paragraphs, therefore, the transfer functions relating the two complexity measures to failure diagnosis time are affected not only by system characteristics and individual differences; rather, the equations are also affected by types of failure-related event, modes of failure diagnosis, and the way in which diagnosis times were aggregated. In the next section, these factors are considered analytically and compared with the transfer functions.

Empirical Analysis. Given the preceding discussion, subject data files were replayed<sup>1</sup> in order to gather failure-related event information. When a subject repaired a failure, it was classified as being topographic, symptomatic, or serendipitous by using the following heuristics. A failure diagnosis was classified as topographic if the failure was affecting the last higher display page viewed by the subject; the assumption was made that the subject was tracing the cause of symptoms via the physical structure of the system. A diagnosis was classified as symptomatic if the subject jumped more than one level in the display hierarchy, if the subject jumped laterally on the same level, or if the subject diagnosed the failure on the basis of contextual messages. All of these instances suggested that the subject was using contextual knowledge of the system to recall the likely location of failures. Finally, a diagnosis was classified as serendipitous if no

---

<sup>1</sup>The comparisons made in this section are limited to data from Session 2. Since the major goal is to show how the results that did arise are explainable, the explanation can be accomplished by examining only a subset of all the data.

symptoms existed on the previous level and for the second, third, etc. failures diagnosed on a single display page.

Failure occurrence and symptom emergence times were also determined. This information was collected for each diagnosis mode (i.e., topographic, symptomatic, and serendipitous) and also aggregated across all diagnosis modes. Using these data, the average time from each event type to the time of diagnosis was calculated. A comparison of these average times to the transfer functions lag values for Session 2 may be found in Table 3. Table 3 may be interpreted as follows. For each subject (1-8) the total number of failures repaired for each diagnosis mode are listed along with the fraction of the total for each mode. The top row in each pair of boxed numbers corresponds to average event times that are approximately equal to lag values from the transfer functions. The lower row in each box contains information about the corresponding transfer function variable. The + or - represents the numerical sign of the transfer function coefficient, "struct" or "strat" refers to the type of complexity, and the final number is the time lag value.

Table 4 presents some of the information in Table 3 in a slightly different form, listing only the empirical average time values paired with transfer function lag values. As Table 4 clearly shows, a very high degree of correlation exists between the time values and the lag values ( $r = 0.92$ ,  $p < 0.01$ ).

Several patterns are evident in Table 3. First, of the eight subjects, seven have transfer function lags that are approximately less than or equal to the overall average time to failure diagnosis (all except Subject 8). Four of these lags involve a strategic complexity component (Subjects 2, 4, 5, and 7), and four involve a structural

component (Subjects 1, 3, 6, and 7 — Subject 7 has both).

Furthermore, the numerical sign of the transfer function coefficient in each case is positive. In each case, therefore, the strategic complexity measure is related positively to the predicted failure diagnosis time.

This finding is intuitively plausible. The measure of strategic complexity reflects the trade-off between the subject's system state uncertainty and the probability of failures existing within the system. If this measure is at a relatively high level when a failure occurs, the time needed to find that failure will be increased due to the number or severity of potential problem areas within the system. A high measure of strategic complexity suggests that many subsystems (clusters) have potential problems and thus, require the attention of the operator. The time necessary to observe these clusters has the cumulative effect of increasing time to failure diagnosis.

Similarly, the measure of structural complexity estimates the minimum number of pages that the subject would have to view in order to find all system failures. On the average, the time needed to locate any one failure in the system will increase as this measure increases.

A second pattern that exists within these results is the similarity between the average time from symptom emergence and the lag values associated with a negative structural complexity component (Subjects 2, 3, 4, 6, and 8). After a symptom emerges, therefore, the structural complexity measure decreases the predicted time to failure diagnosis: the greater the structural complexity of the system, the less time it takes to locate failures. This counter-intuitive result may be explained as follows: as system structural complexity increases, more failures exist within the system. As the number of failures in the system

increases, it is likely that a subject will locate some failures rather quickly.

This observation reflects the relation between fault diagnosis time and number of failures in the system. As the number of failures in the system increases, one might expect the time to diagnosis for some failures also to increase. On the other hand, as the number of failures in the system increases, the chances of finding some failures fairly soon is relatively high. Thus, once a symptom emerges, the average time to failure diagnosis will decrease simply because there are more possible failures in the system to find.

This conclusion is consistent with the relation between lag values and the calculated average time between symptom emergence and failure diagnosis for serendipitous diagnoses. Four of the eight subjects (Subjects 2, 3, 4, and 8) have a negative structural complexity component that is related to the symptom emergence for this mode of failure diagnosis. This increase/decrease effect of complexity, therefore, appears to be dependent upon both the type of complexity (structural or strategic) and the type of failure-related event (e.g., failure occurrence or symptom emergence).

In one situation, the measure of structural complexity at the time of symptom emergence appears to increase failure diagnosis time (Subject 7). It is worthwhile noting that this subject located substantially more failures than any other subject (97 vs. 75 -- the next highest total). This high number was not due to an effective strategy; rather, the subject used a poor strategy that resulted in a very high number of capacity failures -- note the high percent of serendipitous failure locations relative to the other subjects. Indeed, all of the

coefficients in Subject 7's transfer function (Table 4) have positive coefficients. In short, the subject was unable to overcome the number of failures in the system, thereby resulting in an increasing time to failure diagnosis.

So far the discussion has centered on the aggregated mode of failure diagnosis. Examining the individual modes of failure diagnosis, similar trends are apparent except in one case: the transfer function associated with topographic diagnoses have no lag values that correspond to any of the inter-event times. What characteristic of topographic diagnoses could cause this lack of association? One possible reason is simply that there are fewer topographic diagnoses made by subjects; thus, it is less likely to obtain an accurate measure of the true mean event time and transfer function lag. Inaccuracy in the measure obscures the nature of the relationship.

Another possibility is related to the length of time necessary to find topographic failures: in general, it takes subjects more time to identify a failure topographically than some other way. (For example, the average time to failure diagnosis for topographic failures for Session 2 data is 124.95s; for symptomatic failures, 45.07s; and for serendipitous failures, 60.03s.) Because of the longer times (note in particular the time from first symptom emergence to failure diagnosis for each subject), the complexity measure is not related to the lag values. Failure diagnosis time in this case is more dependent upon the probabilistic nature of the queueing network than the skills or thresholds of individual subjects.

Summary. The preceding discussion indicates that the variables and lags present in the transfer functions are reasonable, if not entirely

---

explainable. The real time values of the lags frequently agree with the average inter-failure event times calculated from subject data files. A comparison of these values for Session Two data suggests that certain recurring patterns of agreement exist between the lags and inter-event times. These recurring patterns are useful in terms of explaining the presence of both positive and negative terms in the transfer functions. Differences between time values can probably be accounted for by any of several reasons, including the high variability present within the data, the subjective nature of the modelling process, and the existence of events other than failure occurrence or symptom emergence (e.g., diagnosis time for a particular system level or subsystem) that affect parameters in the transfer functions.

Results reported in this section demonstrate how two different dimensions of complexity, structural and strategic, can be related to human fault diagnosis skills in a large scale system. The exact nature of the two measures is relatively unimportant beyond a certain degree of intuitive validity. The importance of these results, however, lies in the demonstration that the complexity measures are dependent upon the number of failures in the system and the rate at which their symptoms emerge. These factors are highly dependent upon both system characteristics (i.e., number of levels and degree of redundancy) and subject strategy. Of equal importance is the demonstration that the complexity measures relate to performance in a time-varying manner, and the nature of this time-varying manner is highly dependent upon events that occur within the system and the strategy of individual subjects.

## CONCLUSION

The experiment, results, and conclusions in this paper have considered the relationship between the design of a large scale system and human monitoring and control behavior. System characteristics such as number of levels and degree of interconnectedness can have a very strong effect on the ability of humans to maintain proper system operation in the presence of failures. Since normal system operation tends to be affected in the opposite direction in the presence of the same design characteristics, system designers must be careful to create environments that support both system and human performance.

Some rather straightforward measures were used to assess the complexity of a large scale system as it relates to the task of monitoring and control. Complexity, as discussed in this paper, is a dynamic property of a human-machine system. Complexity varies with time and it varies among operators. Furthermore, complexity is multi-dimensional; two dimensions of complexity (i.e., structural and strategic) have been proposed, and it appears that this distinction is useful, both conceptually and practically. Complexity is not due solely to the structure of the system, although a system may certainly be complex due to its structure. Rather, complexity also arises when the human, trying to solve problems within the system's environment, does not understand the structure, and, as a result issues an inappropriate command, misinterprets display information, etc. In short, systems are also complex due to the human's understanding of the system as reflected by his strategy.

Another result from this work concerns the outcome of complexity. Based on a review of the literature and the major control task of subjects (i.e., finding failures), average time to failure diagnosis was



used as the major dependent measure of complexity. As results suggest, however, average time to failure diagnosis alone does not completely describe the implications of complexity. For example, the most complex systems resulted in shorter failure diagnosis times due to the number and location of failures. A smaller fraction of the total number of failures was diagnosed, however. Thus, fraction failures diagnosed was used to explain a different aspect of performance related to task complexity. In short, the result of complexity is multi-dimensional. A single dimension does not capture the outcome of a complex system.

These comments are important in light of the relationships among system characteristics that contribute to complexity, proper operation of the system, and complexity of monitoring and control by the human. As the system becomes more "complex" (from a non-behaviorist's perspective, i.e., more levels and more redundancy), it becomes more resistant to the effects of system failures. Failures take longer to propagate through the more complex systems. Moreover, the effects of any one failure on overall system performance are minimized due to the number of alternate paths through the system. Hence, normal system operation is enhanced. This situation is analogous to the use of redundant or stand-by equipment in systems to increase fault tolerance. On the other hand, as the system becomes more complex, the task of finding system failures becomes more difficult. Although the system design characteristics can help to avoid the short term effects of failures, they can have the dual effect of making the human supervisory controller's task more difficult.

The relationship between complexity and human performance takes on increasing importance given the growing prevalence of large scale systems. Human abilities and limitations in monitoring and controlling

these complex systems must be identified in order to design systems that facilitate good failure diagnosis and network management performance. In short, systems must be designed such that they do not overload human information processing capabilities. Beyond the issue of design, an understanding of human performance constraints should facilitate the creation of effective performance aids. Such aids can be used to help people overcome their limitations in coping with the complex environments these systems create, thereby leading to safe and effective system performance.

#### ACKNOWLEDGEMENT

This work was supported by the Army Research Institute for the Behavioral and Social Sciences under Contract MDA 903-82-C-0145.

## REFERENCES

- [1] J. Rasmussen and W.B. Rouse, eds., Human Detection and Diagnosis of System Failures, New York: Plenum Press, 1981.
- [2] R.L. Henneman and W.B. Rouse, "Human Performance in Monitoring and Controlling Hierarchical Large-Scale Systems," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-14, no. 2, pp. 184-191, March/April 1984.
- [3] S.A. Cook, "An overview of computational complexity," Communications of the ACM, vol. 26, no. 6, pp. 401-408, June 1983.
- [4] J. Hartmanis and J.E. Hopcroft, "An overview of the theory of computational complexity," Journal of the ACM, vol. 18, no. 3, pp. 444-475, July 1971.
- [5] H.R. Lewis and C.H. Papadimitriou, "The efficiency of algorithms," Scientific American, vol. 238, no. 1, pp. 96-109, January 1978.
- [6] R.G. Parker and R.L. Rardin, "An overview of complexity theory in discrete optimizations: Part I. Concepts," IIE Transactions, vol. 14, no. 1, pp. 3-10, 1982a.
- [7] R.G. Parker, and R.L. Rardin, "An overview of complexity theory in discrete optimization: Part II. Results and implications," IIE Transactions, vol. 14, no. 2, pp. 83-89, 1982b.
- [8] S.R. Das, Z. Chen, T. Lin, and C.L. Sheng, "Complexity and performance of a graph theory algorithm for cluster analysis," International Journal of Computer Mathematics, vol. 13, pp. 41-50, 1983.
- [9] B. Borovac, M. Vukobratovic, and D. Stokic, "Analysis of the influence of actuator model complexity on manipulator control synthesis," Mechanism and Machine Theory, vol. 18, no. 2, pp. 113-122, 1983.
- [10] M.L. Fredman, "Observations concerning the complexity of a class of on-line algebraic problems," IEEE Transactions on Computers, vol. C-30, no. 1, pp. 83-86, January 1981.
- [11] R.E. Ladner, "The complexity of problems in systems of communicating sequential processes," Journal of Computer and System Sciences, vol. 21, pp. 179-194, 1980.
- [12] W.B. Rouse and S.H. Rouse, "Measures of complexity of fault diagnosis tasks," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-9, no. 11, pp. 720-727, November 1979.
- [13] H. Fujiwara and K. Kinoshita, "On the computational complexity of system diagnosis," IEEE Transactions on Computers, vol. C-27, no. 10, pp. 881-885, October 1978.

- [14] R.W. Priester and J.B. Clary, "New measures of testability and test complexity for linear analog failure analysis," IEEE Transactions on Computers, vol. C-30, no. 11, pp. 884-888, November 1981.
- [15] S. Henry and D. Kafura, "Software structure metrics based on information flow," IEEE Transactions on Software Engineering, vol. SE-7, no. 3, pp. 510-518, September 1981.
- [16] M.H. Halstead, Elements of software science, New York: Elsevier, 1977.
- [17] N.S. Coulter, "Software science and cognitive psychology," IEEE Transactions on Software Engineering, vol. SE-9, no. 2, pp. 166-171, March 1983.
- [18] V.Y. Shen, S.D. Conte, and H.E. Dunsmore, "Software science revisited: a critical analysis of the theory and its empirical support," IEEE Transactions on Software Engineering, vol. SE-9, no. 2, pp. 155-165, March 1983.
- [19] T.J. McCabe, "A complexity measure," IEEE Transactions on Software Engineering, vol. SE-2, no. 4, pp. 308-320, December 1976.
- [20] J.S. Davis, "An investigation of chunk based complexity measures," Ph.D. dissertation, Georgia Institute of Technology, 1984.
- [21] B.D. Chaudhary and H.V. Sahasrabuddhe, "Two dimensions of program complexity," International Journal of Man-Machine Studies, vol. 18, pp. 505-511, 1983.
- [22] W. Weaver, "Science and Complexity," American Scientist, Autumn, 1948.
- [23] H.A. Simon, Sciences of the artificial, 2nd edition, The MIT Press: Cambridge, MA, 1982.
- [24] W.R. Ashby, An introduction to cybernetics, John Wiley and Sons: New York, 1956.
- [25] C. Alexander, Notes on the synthesis of form, Harvard University Press: Cambridge, MA, 1964.
- [26] R.D. Brunner and G.D. Brewer, Organized complexity, The Free Press: New York, 1971.
- [27] R.J. Waller, "Complexity and the boundaries of human policy making," International Journal of General Systems, vol. 9, pp. 1-11, 1982.
- [28] A.J. Mackinnon and A.J. Wearing, "Complexity and decision making," Behavioral Science, vol. 25, pp. 285-296, 1980.
- [29] I. Bazovsky, Reliability theory and practice, Prentice-Hall, Inc.: Englewood Cliffs, NJ, 1961.

- [30] A.S. Goldman and T.B. Slattery, Maintainability: a major element of system effectiveness, John Wiley and Sons: New York, 1964.
- [31] W.H. von Alven, Reliability engineering, Prentice-Hall, Inc.: Englewood Cliffs, NJ, 1964.
- [32] W.R. Ashby, Design for a brain, John Wiley and Sons: New York, 1960.
- [33] L. Lofgren, "Complexity of descriptions of systems: a foundational study," International Journal of General Systems, vol. 3, pp. 197-214, 1977.
- [34] R. Rosen, "Complexity as a system property," International Journal of General Systems, vol. 3, pp. 227-232, 1977.
- [35] P.T. Saunders and M.H. Ho, "On the increase in complexity in evolution II. The relativity of complexity and the principle of minimum increase," Journal of Theoretical Biology, vol. 90, pp. 515-530, 1981.
- [36] S.N. Greenberg and L.E. Krueger, "Effect of letter orientation and sequential redundancy on the speed of letter search," Memory and Cognition, vol. 11, no. 2, pp. 181-191, 1983.
- [37] M.H. Bornstein and M.D. Monroe, "Chromatic information processing: rate depends on stimulus location in the category and psychological complexity," Psychological Research, vol. 42, pp. 213-225, 1980.
- [38] G.R. Dirkin, "Cognitive tunneling: use of visual information under stress," Perceptual and Motor Skills, vol. 53, pp. 191-198, 1983.
- [39] H.L. Dee and H.J. Hannay, "Reversal of asymmetry in human perceptual performance as a function of labeling, mode of response, and familiarity," Perceptual and Motor Skills, vol. 52, pp. 183-193, 1981.
- [40] J. Hochberg and V. Brooks, "The psychophysics of form: reversible-perspective drawings of spatial objects," The American Journal of Psychology, vol. 73, no. 3, p. 337-354, September 1960.
- [41] P.C. Vitz and T.C. Todd, "A model of the perception of simple geometric figures," Psychological Review, vol. 78, no. 3, pp. 207-228, 1971.
- [42] D.L. Butler, "Predicting the perception of three-dimensional objects from the geometrical information in drawings," Journal of Experimental Psychology, vol. 8, no. 5, pp. 674-692, 1982.
- [43] F. Attneave, "Physical determinants of the judged complexity of shapes," Journal of Experimental Psychology, vol. 53, no. 4, pp. 221-227, April 1957.

- [44] R. Kimchi and S.E. Palmer, "Form and texture in hierarchically constructed patterns," Journal of Experimental Psychology: Human Perception and Performance, vol. 8, no. 4, pp. 521-535, 1982.
- [45] H.A. Simon, "Complexity and the representation of patterned sequences of symbols," Psychological Review, vol. 79, no. 5, pp. 369-382, September 1972.
- [46] J.B. Brooke and K.D. Duncan, "Effects of system display format on performance in a fault location task," Ergonomics, vol. 24, no. 3, pp. 175-189, 1981.
- [47] J.A. Glover, B.S. Plake, and J.W. Zimmer, "Distinctiveness of encoding and memory for learning tasks," Journal of Educational Psychology, vol. 74, no. 2, pp. 189-198, 1982.
- [48] M.A. McDaniel, "Syntactic complexity and elaborative processing," Memory and Cognition, vol. 9, no. 5, pp. 487-495, 1981.
- [49] M.H. Ashcraft and E.H. Stazyk, "Mental addition: a test of three verification models," Memory and Cognition, vol. 9, no. 2, pp. 185-196, 1981.
- [50] E.F. Loftus and P. Suppes, "Structural variables that determine problem-solving difficulty in computer-assisted instruction," Journal of Educational Psychology, vol. 63, no. 6, pp. 531-542, 1972.
- [51] B.B. Morgan and E.A. Alluisi, "Acquisition and performance of a problem-solving skill," Perceptual and Motor Skills, vol. 33, pp. 515-523, 1971.
- [52] D.E. Kieras and P.G. Polson, "An approach to the formal analysis of user complexity," International Journal of Man-Machine Studies, in press.
- [53] J.G. Wohl, "Maintainability prediction revisited: diagnostic behavior, system complexity, and repair time," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-12, no. 3, pp. 241-250, 1982.
- [54] J.G. Wohl, "Cognitive capability vs. equipment complexity in electronic maintenance," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-13, no. 4, 1983a.
- [55] J.G. Wohl, J.G., "Connectivity as a measure of problem complexity in failure diagnosis," Proceedings of the Human Factors Society - 27th Annual Meeting, pp. 681-684, 1983b.
- [56] R.L. Henneman, "Human problem solving in complex hierarchical large scale systems," Ph.D. thesis, Georgia Institute of Technology, January 1985.

- [57] G.E.P. Box and G.M. Jenkins, Time series analysis: forecasting and control, Holden-Day: San Francisco, 1976.
- [58] D.C. Montgomery and G. Weatherby, "Modeling and forecasting time series using transfer function and intervention methods," AIIE Transactions, vol. 12, no. 4, pp. 289-306, 1980.

### List of Figures

- Figure 1     CAIN Display
- Figure 2     Example calculation of structural complexity
- Figure 3     Example monitor and cluster display for calculation of strategic complexity
- Figure 4     Summary of results from analysis of variance of complexity measures
- Figure 5     Summary of results from correlation analysis

### List of Tables

- Table 1       Summary of CAIN Commands
- Table 2       Example calculation of strategic complexity
- Table 3       Summary of average times from failure-related events to failure diagnosis (Session 2)
- Table 4       Summary of average event times and lag values



Time = 523.6

1	3	5	3	9	3	13	
Seattle		Minneapol		Chicago		Boston	
2		6	4	10	3	14	
SanFranci		Denver		Cincinnati		NewYorkCi	
3	2	7	2	11		15	1
LosAngele		KansasCit		Atlanta		Washington	
4	1	8	8	12		16	
Phoenix		Dallas		NewOrlean		Miami	

A: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16  
B: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16  
C: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Your action:

Figure 1 CAIN Display

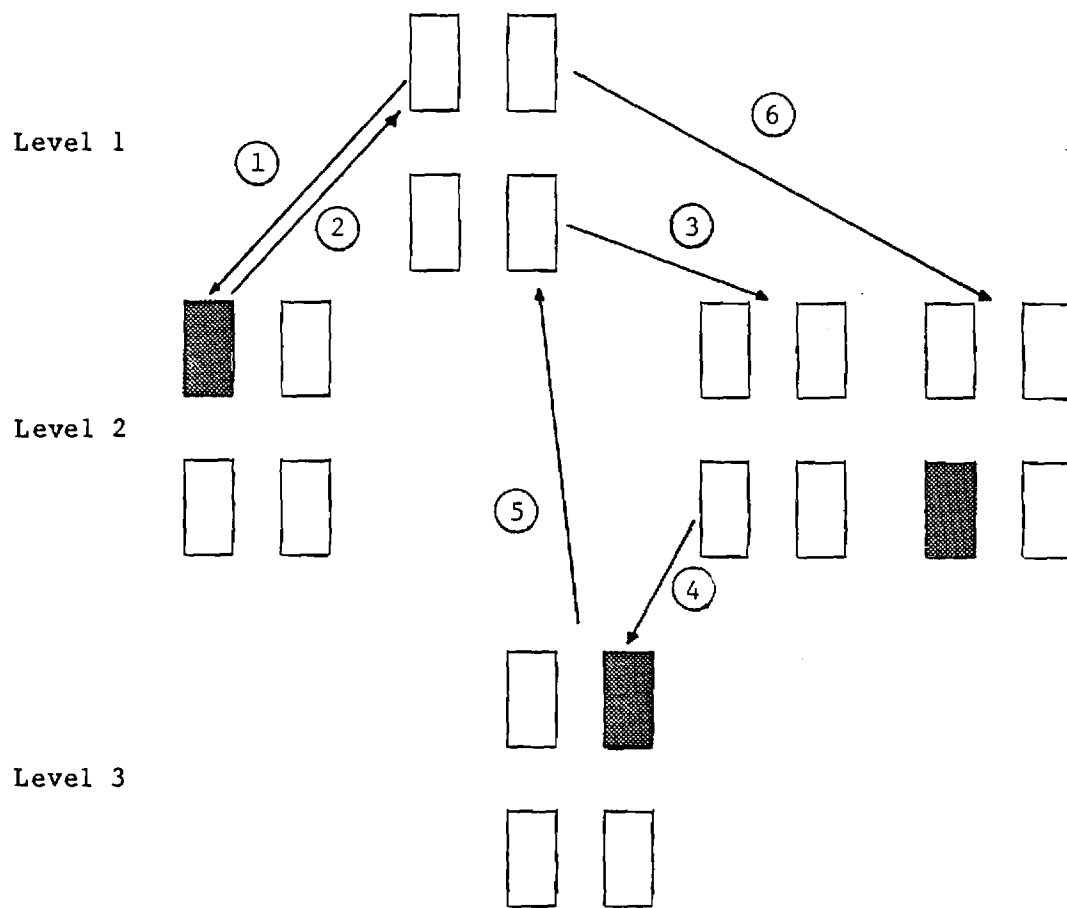


Figure 2 Example calculation of structural complexity

Monitor Display

<u>Cluster</u>	<u>Number of customers</u>
Denver	8
Los Angeles	5
Chicago	1
New York	1

Cluster Display

<div>15</div>	<div>1</div>
Denver	Chicago
<div>2</div>	<div>5</div>
Los Angeles	New York

Figure 3 Example monitor and cluster display for calculation of strategic complexity

		REDUNDANCY	
		HIGH	LOW
NO. OF LEVELS	2	LOW Structural Complexity MODERATE Strategic Complexity	MODERATE Structural Complexity LOW Strategic Complexity
	3	MODERATE Structural Complexity HIGH Strategic Complexity	HIGH Structural Complexity LOW Strategic Complexity

Figure 4 Summary of results from analysis of variance  
of complexity measures

		Fraction Failures Diagnosed	Avg. Time to Failure Diagnosis
Structural C	2	-	
	3	-	-
Strategic C	2	-	
	3	+	

Figure 5 Summary of results from correlation analysis

Table 1 Summary of CAIN commands

ACCESS Commands

dCITY	down CITY
u	up one level
fCITY	find CITY
a	return to top level

MONITOR Commands

m	monitor
s	system statistics
w	watch list
+CITY	add CITY to watch list
-CITY	delete CITY from watch list
o	list repair orders
e	erase warning message from bottom of screen

DIAGNOSTIC Commands

t	tests displayed cluster of nodes
cCITY	information about CITY

CONTROL Commands

rCITY	replaces equipment in CITY
lCITY=%load	alters load in cluster CITY to %load
lsys=%load	alters load in entire system
l	displays load

Table 2 Example calculation of strategic complexity

<u>Cluster</u>	<u>U</u>	<u>p[F X]</u>		<u>U x p[F X]</u>
	<u>uncertainty</u>	<u>monitor</u>	<u>cluster</u>	
Denver	20.12	.600	.750	15.090
Los Angeles	0.54	.100	.015	.054
Chicago	7.36	.001	.001	.007
New York	9.12	.001	.050	<u>.456</u>
Strategic Complexity =				15.607

Table 3 Summary of average times from failure-related events to failure diagnosis (Session 2)

	All	Topographic	Symptomatic	Serendipitous
<u>Subject 1</u>				
Total Failures	58	20	15	23
Frac. of Total		0.35	0.26	0.40
$\bar{T}(\text{Failure})$	63.02 + struct 48.1	81.68	27.47	69.99 + struct 48.1
$\bar{T}(\text{Symptom})$	29.65	38.18	14.98	31.81
<u>Subject 2</u>				
Total Failures	61	10	18	33
Frac. of Total		0.16	0.30	0.54
$\bar{T}(\text{Failure})$	75.85 + strat 71.1	101.03	47.66	83.59
$\bar{T}(\text{Symptom})$	32.29 - struct 22.4	42.99	24.44 - struct 22.4	33.33 - struct 22.4
<u>Subject 3</u>				
Total Failures	66	16	17	33
Frac. of Total		0.24	0.26	0.50
$\bar{T}(\text{Failure})$	78.04 + struct 60.1	180.32	42.52 - struct 40.1	46.75 - struct 40.1
$\bar{T}(\text{Symptom})$	44.29 - struct 40.1	97.88	19.57 + struct 10 - strat 16.7	31.03 - struct 23.4
<u>Subject 4</u>				
Total Failures	63	22	19	22
Frac. of Total		0.35	0.30	0.35
$\bar{T}(\text{Failure})$	53.90 + strat 46.4	78.23	34.06 + strat 23.2	46.71 + strat 46.4
$\bar{T}(\text{Symptom})$	31.80 - struct 11.6	47.56	14.45 - struct 11.6	31.03 - struct 11.6



Subject 5

Total Failures	75	16	23	36
Frac. of Total		0.21	0.31	0.48

$\bar{T}$ (Failure)	66.93 + strat 67.6	146.20	50.90	41.93
---------------------	-----------------------	--------	-------	-------

$\bar{T}$ (Symptom)	28.47	56.91	19.98 + strat 11.6	21.25 + strat 11.6
---------------------	-------	-------	-----------------------	-----------------------

Subject 6

Total Failures	43	16	20	7
Frac. of Total		0.37	0.47	0.16

$\bar{T}$ (Failure)	64.30 + struct 66.2	94.48	39.31	66.75
---------------------	------------------------	-------	-------	-------

$\bar{T}$ (Symptom)	31.01 - struct 29.4	52.16	10.44	41.43
---------------------	------------------------	-------	-------	-------

Subject 7

Total Failures	97	17	15	65
Frac. of Total		0.18	0.16	0.67

$\bar{T}$ (Failure)	78.89 + struct 49.8 + strat 49.8	143.59	57.35 + struct 49.8 + strat 49.8	66.94
---------------------	----------------------------------------	--------	----------------------------------------	-------

$\bar{T}$ (Symptom)	45.74	88.15	26.42	39.11
---------------------	-------	-------	-------	-------

Subject 8

Total Failures	47	12	8	27
Frac. of Total		0.26	0.17	0.57

$\bar{T}$ (Failure)	103.95	174.07	61.32	85.42
---------------------	--------	--------	-------	-------

$\bar{T}$ (Symptom)	65.01 - struct 44.2	116.95	24.69	53.88 - struct 44.2
---------------------	------------------------	--------	-------	------------------------

Table 4 Summary of average event times and lag values

<u>Average time</u>	<u>lag value</u>	<u>Average time</u>	<u>lag value</u>
63.0	48.1	19.6	10.0
75.9	71.1	19.6	16.7
32.3	22.4	34.1	23.2
78.0	60.1	14.5	11.6
44.3	40.1	20.0	11.6
53.9	46.4	57.4	49.8
31.8	11.6	70.0	48.1
66.9	67.6	33.3	22.4
64.3	66.2	46.8	40.1
31.0	29.4	31.0	23.4
78.9	49.8	46.7	46.4
65.0	44.2	31.0	11.6
24.4	22.4	21.3	11.6
42.5	40.1	53.9	44.2

A MODEL OF HUMAN PERFORMANCE  
IN A  
LARGE SCALE DYNAMIC SYSTEM

Richard L. Henneman

Center for Man-Machine Systems Research  
School of Industrial and Systems Engineering  
Georgia Institute of Technology  
Atlanta, GA 30322

**Abstract**

A model of human performance in monitoring and controlling complex engineering systems is considered from the perspective of implementing the model as an on-line performance aid. Results from the literature are discussed in the context of CAIN, a simulated large scale system that has been used to study human supervisory control performance [1,2]. A rule-based model of human performance in CAIN is proposed and a methodology for evaluation is suggested.

**Introduction**

Recent trends in automation have facilitated the creation of large, complex engineering systems. Due to the capabilities of computer technology to control a large number of interconnected components, the normal operation of these systems is typically left to an automatic controller. During unforeseen events (such as system failures) that cannot be handled by the computer, however, a human controller must take corrective action. As many have noted [3], this increase in automation is fundamentally changing ways in which people interact with large systems. The human operator no longer is in charge of the routine, continuous control of the system. Rather, the operator is mostly concerned with the unexpected, the unusual, and the non-routine aspects of system control. Requisite human skills for system control are shifting from psychomotor to problem solving [4].

Recent research activity has focused on human supervisory control and problem solving in complex engineering environments [3], although a majority of this work has been restricted to the process control domain. Henneman and Rouse [1,2] discuss human performance in monitoring and controlling a large scale dynamic network (such as a communication network). These systems can be represented as discrete queueing networks. Henneman and Rouse [1,2] have conducted a series of experiments that empirically assessed the effects of system features (such as number of levels and display size) on human performance via two simulated large scale systems called MABEL and CAIN. Other efforts have been directed towards developing and evaluating measures of large scale system complexity. Results to this point have led to a good empirical understanding of the relationship between the structure of the system and human performance.

Given this understanding of human performance in this task, it is now possible to postulate a model of human problem solving in the CAIN environment. Thoughts presented in this paper are directed towards the development of such a model. First, some previous related modeling efforts are reviewed. Second the CAIN environment is briefly described. Finally, the problem solving model is presented and discussed.

**Background**

The model developed in this paper is an extension of a conceptual model of human problem solving proposed by Rouse [5]. Rouse has suggested that problem solving takes place on three levels: 1) recognition and classification, 2) planning, and 3) execution and monitoring. Thus, when a problem situation develops, the first task is to detect that the problem exists and to categorize it (recognition and classification). An approach or plan to solving the problem must then be developed (planning), and finally, the plan must be implemented (execution and monitoring). The model is further characterized by its ability to make either a state- or a structure-oriented response, depending on both the system state and the human's level of expertise. The model assumes that humans have a preference for pattern-recognition solutions to problems -- that is, humans prefer to make context-specific state oriented responses to situations. Moreover, the model operates heterarchically at all three problem solving levels almost simultaneously, with situations constantly being re-evaluated relative to their state- or structure-oriented status.

The model as presented by Rouse [5] is explicitly a conceptual realization/ combination of other more restricted problem solving models. Knaeuper and Rouse [6] attempted to implement an operational model of this conceptual framework in a computer simulated process plant environment called PLANT [7]. They developed a rule-based model called KARL (Knowledgeable Application of Rule-based Logic) that controlled the computer simulated process plant. KARL consists of a set of production rules that comprise the knowledge base and a control structure that accesses that knowledge base. KARL's structure is

defined by the three levels of problem solving described above and also four major tasks that are associated with human performance in a process control environment (i.e., transition, steady-state tuning, failure detection and diagnosis, and failure compensation). Thus, changes were made to the originally proposed model in order to accommodate specific characteristics of process plants. In addition, the model does not explicitly incorporate a mechanism to distinguish between state- and structure-oriented responses.

KARL's performance in controlling PLANT was compared with that of human subjects. Overall, the comparison was favorable in terms of such performance measures as plant output and plant stability. An action-by-action comparison between KARL and subjects revealed, however, two major systematic differences: first, subjects tended to be more conservative in terms of selecting levels of system input and output, and second, KARL tended to adjust input and output more frequently than subjects. These findings were probably a result of differences between subjects' underlying performance goals and KARL's goals. KARL possessed mechanisms that always tried to maximize plant production, a strategy which it, unlike subjects, pursued inflexibly. Consequently, KARL tended to be more extreme in terms of accurately following procedures.

Knaeuper and Morris [8] attempted to use KARL as an on-line aid to subjects controlling PLANT. In light of the difficulties PLANT subjects had in accurately assessing situations and following appropriate procedures [9] and since KARL was good at these activities, the use of KARL as an on-line aid was a logical extension. KARL provided three types of aid: 1) situation assessment (i.e., identification of the appropriate procedure), 2) guidance in following procedures, and 3) performance feedback. Comparing performance of subjects who received help from KARL to those who performed unaided, the aided subjects maintained a higher level of plant stability, scored higher on a paper-and-pencil test of system knowledge, and were more successful in diagnosing an unfamiliar system failure.

As Knaeuper and Morris [8] indicate, the interpretation of the results is not straightforward. In fact, they conclude that although this experiment successfully demonstrated the viability of the use of a model-based performance aid, issues related to on-line training and aiding are far from resolved. The framework outlined in this paper is an attempt to further investigate the use of a model-based performance aid in a different task domain. The next section, therefore, describes a simulated large scale system used to study human failure diagnosis performance.

#### CAIN: A Simulated Large Scale System

Two previous experiments [1,2] have considered human performance in the monitoring and control of a computer simulated large scale system. In the first experiment,

subjects supervised an essentially context-free representation of a large scale network called MABEL (Monitoring, Accessing, Browsing, and Evaluating Limits), trying to optimize such system parameters as number of customers served and customer processing time while trying to diagnose system failures. In the second experiment, the MABEL scenario was substantially augmented to produce a higher fidelity system. This new scenario is called CAIN (Contextually Augmented Integrated Network). The remainder of this section provides a brief overview of CAIN. The reader is referred to Henneman and Rouse [2] for more detail.

#### Overview of CAIN

CAIN is programmed in Pascal on a VAX 11/780 computer and operates in real time. It is structured as a large hierarchical network that can range in size from hundreds to thousands of nodes. Customers travel through the system from a randomly selected source node to a random destination. Subjects monitor this system activity via a CRT display. When they detect a problem in the system (possibly due to a failure), subjects issue an appropriate command through a keyboard to correct and compensate for the abnormal situation. The overall objectives of the operator are: 1) to maximize the number of customers served, and 2) to minimize customer sojourn time.

Because of the network size, it is not possible to display information about all nodes at one time. Thus, nodes are grouped into relatively small networks called clusters. Human operators are restricted to viewing only one cluster at a time on the CAIN display. Clusters are grouped into hierarchic levels.

#### Effects of node failures

Under normal circumstances, CAIN operates automatically without human intervention. Since the system cannot automatically diagnose and repair failures, the human must monitor the system looking for evidence of failed components. Node failures can occur in two ways. The first is a randomly occurring failure mode caused by malfunctioning equipment. The second type, capacity failure, can be caused by the randomly occurring failures. Each node has a maximum number of customers that it can store at a time. If this limit is exceeded, the node fails. Thus, if a node fails randomly and a customer needs to visit that node, it will be retained at its previous node. This retention will cause the previous node to stop processing customers, which can lead to a capacity failure. In this way, if the operator does not locate failures quickly, the problems will propagate through the system.

#### Addition of context

Although the physical hierarchical structure of MABEL was preserved, the addition of contextual information to CAIN required changing both interface and system characteristics. In CAIN, for example, each node in the system is identified by a

specific geographic location (for example, nodes in the highest level of the system are labelled Seattle, Chicago, Miami, etc.) In addition, the contextual fidelity was enhanced through the addition of associative links (i.e., memory aids) and cues (i.e., clues to the location of system problems). Associative links were formed by requiring subjects to reference nodes via their geographic label. Cues were formed by the introduction of context-dependent events, such as recurring failures and non-uniform loading.

#### A Model of Human Performance in CAIN

Building from the work of Rouse, Knaeuper and Morris [5,6,8], a model is proposed in this section with the intent of supporting human performance in monitoring and controlling CAIN. First, some overall requirements of the model are specified. Second, a specific model is proposed, and finally, the way in which the model can be used as a performance aid will be discussed.

#### Overall requirements

Before the model is proposed, several requirements that the model should meet are specified in this section. For example, in order to function as a performance aid, the model must be able to represent several different performance strategies. A result from Henneman and Rouse [2] indicated that subjects discovered failures in CAIN using three modes of failure diagnosis: symptomatic, topographic, and serendipitous. Subjects using a topographic strategy trace failure symptoms from higher system levels to lower level causes. Subjects using a symptomatic strategy make a direct mapping from their system structure knowledge to the failed component. A symptomatic diagnosis relies, therefore, on the subject's contextual knowledge of the system. Finally, subjects may also identify failures accidentally or serendipitously. When using this diagnosis mode, subjects locate failures while browsing through the system or while tracing the cause of a different failure.

These failure diagnosis modes are dependent upon an individual subject's understanding of how the system operates as well as the subject's knowledge of the contextual relationship between system components. Therefore, the model should incorporate an explicit representation of both contextual knowledge and task knowledge. Moreover, the model should allow this contextual knowledge to be augmented over time as subjects gain performance expertise.

Finally, the model should represent the way in which subjects prioritize sub-tasks in monitoring and controlling the network. Multiple system failures and the dynamic nature of the system may cause the operator to have several sub-tasks to perform at any one time. At one instant, for example, the system may have multiple failure symptoms on the display, heavy customer demands in one part of the system, and a failed node. The relative importance of each of these sub-tasks can vary with time. In some cases there will

be clear-cut choices among alternatives while in other cases there will be indifference. It may be argued that the essence of good performance in this task is the subject's ability to prioritize sub-tasks that are present concurrently. An important goal of this model, therefore, is to represent sub-task prioritization.

In summary, underlying the development of this model is the demonstration of how a general representation of human problem solving [5] can be adapted to model human performance in a complex large scale environment. More importantly, this model is to be used as an on-line performance aid. Unlike the work of Knaeuper and Morris [8] in which a performance model was adapted *post hoc* to serve as an aid, the development of this model is motivated by the desire to use it as an on-line aid. In order to achieve these underlying goals, the model should flexibly support several performance strategies, explicitly represent contextual knowledge, and contain a mechanism to prioritize sub-tasks.

#### A model

A model that meets these requirements is shown in Figure 1. As in Knaeuper and Rouse's KARL [6], the model proposed here will be represented as a set of if-then rules organized into a hierarchical structure. The model contains two levels of activities. The lowest level of the model consists of the three stages of problem solving discussed by Rouse [5]: recognition/classification, planning, and execution. Because multiple sub-tasks may concurrently exist, the model can be operating in any of these stages. Thus, the highest level of the model contains a mechanism to prioritize the performance of sub-tasks in the three lower level stages. The remaining model component represents the

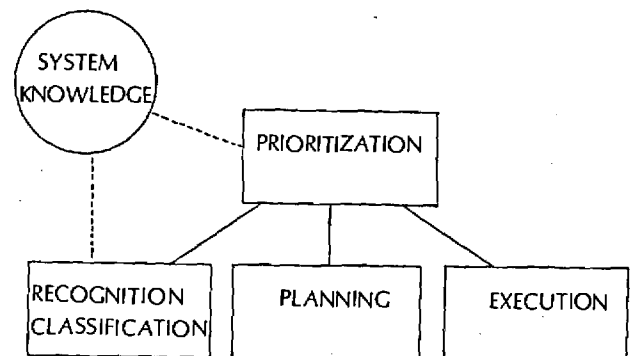


Figure 1. A model of human performance in CAIN

system (or contextual) knowledge needed to perform the task. The following paragraphs will discuss each part of the model in more detail.

Recognition/classification takes place when the subject identifies that an event has occurred or a situation exists. Examples include node failure, failure symptoms, abnormal customer demands, and normal situations.

Once an event has been recognized and classified, the subject must develop an approach to improve the situation. A difference exists between planning at this level and the prioritization or the coordination of low level plans that occurs at the highest model level. Plans at this low level are best compared to simple scripts [10] or short sequences of actions. To illustrate, consider a situation in which a subject observes an increasing queue size in a node. This event suggests that a failure exists in a lower system level; a suitable plan of action would be to 1) display the lower level cluster, and 2) test the new cluster for failed components.

After a suitable course of action is identified, the plan must be implemented. An analysis of the timing of subject's commands [11] indicated that these command sequences are frequently issued in rapid succession, suggesting that the plans are executed automatically with little conscious attention. In the execution phase of this model these command sequences are issued. The assumption is made that once the sequence is started, it cannot be interrupted.

These three phases - recognition/ classification, planning and execution - form the basis of activities implemented by the model. In general, the performance of each sub-task will progress sequentially through each of the three stages. Nevertheless, since multiple sub-tasks may exist, it is likely that this sequential process may be interrupted by a new sub-task of greater importance. As mentioned previously, the key to good performance in this task is the ability to prioritize sub-tasks. Thus, perhaps the most important feature of this model is the way in which activities are prioritized. Prioritization takes place in the highest model level.

Two other components are included in the model, one of which is implicitly embedded within other model components, the other of which is explicitly represented. These two parts represent the knowledge necessary to perform the task: task knowledge and system knowledge.

Task knowledge (analogous to Anderson's procedural knowledge [12]) encompasses the knowledge of how to do things, for example, how to diagnose a failed component. This knowledge will be embedded in the productions (or if-then rules) associated with the model components (i.e., prioritization, recognition/classification, and planning).

System knowledge (analogous to Anderson's declarative knowledge [12]) encompasses the knowledge of contextual relationships among system components. For example, system knowledge might contain a fact like "Evanston is a second level city that is associated with Chicago." In addition to this static knowledge of system structure, system knowledge also encompasses facts that are related to the system dynamics. For example, the names of nodes with recurring problems or regions with high customer loading are likely to be remembered by subjects. In this model, these facts will be stored as system knowledge. This type of knowledge should only be accessed by the recognition/classification and prioritization model components. Planning and execution are performed independently of contextual knowledge. Methods of representing this task system knowledge are currently being investigated [13], along with ways in which the system knowledge can be augmented as subjects gain more expertise.

#### The model as an aid

This section considers each of the model components and the roles they could play in providing performance assistance. Recognition/classification, for example, will indicate "trouble spots", i.e., regions with a greater likelihood of having a failure or heavy loading. When scanning a display, a subject can easily miss a salient cue. The model should be helpful in terms of indicating those cues that have the greatest likelihood of reflecting a failure.

The planning module can assist by telling operators what to do once they have recognized a situation. This information would be most useful for novice operators. Nevertheless, for some situations that are seen only infrequently, this advice would be useful for all operators. This lack of emphasis on procedural information is markedly different from the advice given by KARL to PLANT subjects [8]. A large part of the advice that KARL provided was procedural information.

Probably the most important aid that the model can offer is in prioritizing sub-tasks. Certain situations are more critical than others; the model should be useful in identifying those sub-tasks that are most important.

Other ways in which the aid should be used is by giving performance feedback and contextual information. As in the KARL-PLANT experiment [8], subjects should receive feedback relative to the success of their actions. In addition, since operators' system knowledge is inevitably at various levels of completeness, the model should assist in augmenting the deficiencies.

The biggest problem in using a model like the one proposed in this paper as an aid is the development of an effective interface between the model and the operator. Since the system state is constantly changing, the information provided by the aid will also be constantly changing. Advice that is relevant

at one time may be erroneous after a few moments. At this time, it is unclear how to control the display of this information. Moreover, since the current CAIN display is already very crowded with verbal information, the addition of advice from a model-based aid may only serve to degrade performance by overloading the human's information processing capabilities. Perhaps synthesized voice output would be a suitable means of presenting this advice.

#### Future work

On-line implementation of the model proposed in this paper has just started. Future plans include the experimental evaluation of the model as an on-line performance aid. The proposed experiment should compare the task performance of two groups of subjects, one of which performs without the aid and the other with the aid. If the aid proves to be successful, another experiment will evaluate the usefulness of individual model components of the aiding scheme.

#### Acknowledgement

This work was supported by the Army Research Institute for the Behavioral and Social Sciences under Contract MDA 903-82-C-0145.

#### References

- (1) R.L. Henneman and W.B. Rouse. "Human performance in monitoring and controlling hierarchical large-scale systems." IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-14, no. 2, pp. 184-191, March/April 1984.
- (2) R.L. Henneman and W.B. Rouse. "On measuring the complexity of monitoring and controlling large scale systems," submitted for publication.
- (3) J. Rasmussen and W.B. Rouse, eds., Human Detection and Diagnosis of System Failures. New York: Plenum Press, 1981.
- (4) C D Wickens, Engineering Psychology and Human Performance, Columbus: Charles E. Merrill, 1984.
- (5) W.B. Rouse, "Models of human problem solving: detection, diagnosis, and compensation for system failures," Automatica, vol. 19, no. 6, pp. 613-625, November 1982.
- (6) A. Knaeuper and W B. Rouse. "A rule-based model of human problem solving behavior in dynamic environments." IEEE Transactions on Systems, Man, and Cybernetics, vol. 15, no. 6, November/December 1985.
- (7) N.M. Morris, W.B. Rouse and J.L. Fath, "PLANT: an experimental task for the study of human problem solving in process control," IEEE Transactions on Systems, Man, and Cybernetics, vol. 15, no. 6, November/December 1985.
- (8) A. Knaeuper and N.M. Morris. "A model-based approach for online aiding and training in process control," in Proceedings of the 1984 IEEE International Conference on Systems, Man, and Cybernetics, Halifax, NS, pp 173-177, 1984
- (9) N.M. Morris and W.B. Rouse. "The effect of type of knowledge on human problem solving in a process control task," IEEE Transactions on Systems, Man, and Cybernetics, vol. 15, no. 6, November/December 1985
- (10) R.C. Schank and R.P. Abelson, Scripts Plans, Goals, and Understanding, Hillsdale, NJ: Lawrence Erlbaum, 1977.
- (11) R.L. Henneman. "Human problem solving in complex hierarchical large scale systems," Ph.D. thesis. Georgia Institute of Technology, January 1985.
- (12) J.R. Anderson. Language, Memory, and Thought, Hillsdale, NJ: Lawrence Erlbaum, 1976.
- (13) R.C. Andes. "Effects of contextual knowledge on human problem solving in large scale systems," M.S. thesis in progress, Georgia Institute of Technology.

**HUMAN PROBLEM SOLVING IN  
DYNAMIC ENVIRONMENTS:  
UNDERSTANDING AND SUPPORTING OPERATORS  
IN LARGE-SCALE, COMPLEX SYSTEMS**

Richard L. Henneman  
William B. Rouse

**FINAL REPORT**

for

Army Research Institute  
Contract No. MDA 903-82-C-0145  
(June 1, 1982 - May 31, 1986)

Center for Man-Machine Systems Research  
Georgia Institute of Technology  
Atlanta, Georgia 30332  
(404-894-3996)

December 1986



HUMAN PROBLEM SOLVING IN DYNAMIC ENVIRONMENTS:  
UNDERSTANDING AND SUPPORTING OPERATORS IN  
LARGE-SCALE, COMPLEX SYSTEMS

Richard L. Henneman  
William B. Rouse

ABSTRACT

Human performance in monitoring and controlling large-scale, dynamic systems is considered. Initial efforts were directed at obtaining an empirical understanding of the relationship between the physical characteristics of a large-scale system and human performance. Results showed the very strong effect that number of levels and degree of interconnectedness can have on human performance. Later efforts formalized these empirical results into several measures of large-scale system complexity. Two dimensions of complexity were proposed, measured, and evaluated: structural complexity results from the physical structure of the system, while strategic complexity results from an operator's understanding of the system. The knowledge gained from these engineering approaches was used to develop a behavioral model of the human operator in a large-scale environment. A comparison of the model's performance to human performance indicated that the model was consistent with human behavior. The model was then used to provide cognitively plausible decision aid to the human operator. Results from this approach were promising in that they showed subtle performance improvement for the aided subjects.

## I. INTRODUCTION

Recent trends in automation have facilitated the creation of large, complex engineering systems, such as communication networks [Henneman and Rouse 1984b, 1986]. These systems are frequently represented as hierarchical networks that consist of a very large number of nodes connected by arcs. The network size and degree of integration combine to create systems of potentially enormous complexity. Because of this complexity, computer technology is often used to control the systems. For example, the functions of a large-scale command and control network may depend on a distributed set of intelligent control systems.

Although full automation often is appropriate during normal situations, control is likely to be transferred to a human operator during abnormal or infrequent events, e.g., system failures. In this way, the manner in which people interact with large systems is fundamentally being changed. The human operator is no longer in charge of the routine, continuous control of the system. Rather, the operator is mostly concerned with the unexpected, the unusual, and the non-routine aspects of system control. Requisite human skills for system control are shifting from psychomotor to problem solving [Wickens 1984].

The use of automation in control systems during normal operation raises questions about the human's ability to control the system during abnormal situations. Since operators infrequently interact with the system, their knowledge of the system dynamics, structure, and context may be inadequate to cope with the complex task demands of abnormal situations. This problem is especially critical in dynamic environments in which the state of the system is time-varying; timely resolution of crisis situations is dependent upon the

human's ability to retrieve and use relevant task knowledge quickly. In addition, in a context-rich environment the human's problem solving performance is also dependent on the human's internal representation of that contextual knowledge.

This report considers human performance in monitoring and controlling large-scale systems by reporting the methods, results, and conclusions of a series of four experiments within a particular environment. Initial efforts were directed at obtaining an empirical understanding of the relationship between the characteristics of a large-scale system and human performance. Later efforts formalized these empirical results into several measures of large-scale system complexity. Finally, the knowledge gained from these engineering approaches was used to develop a behavioral model of the human operator in a large-scale dynamic environment. This model formed the basis of an approach to aiding the human operator. The report roughly follows this chronology.

## II. SYSTEM CHARACTERISTICS AND HUMAN PERFORMANCE

### A. Characteristics Of Large-Scale Systems

When considering human performance in interacting with a complex system, difficulty often arises in trying to exercise adequate experimental control over the characteristics of the real system. A variety of exogenous variables may mask the true effect of the variable of interest. Moreover, due to cost constraints, it is often difficult to make the changes in system characteristics necessary to elicit variations in human performance. For example, if in a large scale system the variable of interest is the number of levels in the system, it is infeasible to alter the structure of a real system. Thus, a common approach to studying human performance in interacting with a complex system is to use a computer-based simulated abstraction of the real system.

Two simulations, MABEL and CAIN, were developed for the purposes of this study. MABEL is a relatively context-free representation of a large scale system in which a human operator is required to monitor and control the real time functioning of the system. By issuing commands, the operator accesses and displays activities within various parts of the hierarchical system, acquires information about the current system state, and issues control actions (e.g., component repairs and load shedding) when required. A subject's major task in MABEL is to diagnose and repair failed components. The CAIN system is structurally isomorphic to MABEL; the difference is that CAIN is contextually augmented, representing the nationwide telephone system.

In order to provide a context for the simulations, experiments, and results presented in this report, the next section describes the general features of two existing large-scale systems, the nationwide telephone system

of AT&T and the U.S. Army's Communications-Electronics Management System. The purpose is to present physical characteristics of these real systems and to describe ways in which people interact with the systems.

## 1. Examples of Large-Scale Systems

The nationwide telephone system [AT&T; Ash and Mummert 1984; Mocenigo and Tow 1984] has functioned until recently as a five-level hierarchical network composed of more than 170 million telephones and more than 22000 switching centers. The network consists of two basic elements: transmission and switching. The transmission elements are the actual communication paths that messages take through the system; the switching stations serve to interconnect calls economically.

A major feature of the system is its high degree of automation. Messages are sent through the system hierarchy via direct or alternate paths that have been pre-determined. The system operates under normal conditions without any manual intervention. The switching stations, serving as repositories of network intelligence, automatically perform such tasks as 1) determination of source, destination, and path through the network; 2) testing of lines for busy conditions before establishing a path; and 3) continual checking of circuit conditions.

Nonetheless, human monitoring and maintaining of the system is still a necessity. During overload situations or in the case of major equipment failures, network performance can degrade rapidly. Human network controllers must intervene when the automatic solutions are excessively expensive or when a problem arises requiring human judgement. To deal with these situations, two general categories of control exist -- expansive and protective [Ash and

Mummert 1984]. Expansive controls increase the network capacity by providing substitute or alternate routes for calls that are blocked. Protective controls reduce the number of calls entering a congested portion of the network or reduce the number of routing alternatives. Thus, the human operator can implement these controls by cancelling alternate routing, rerouting calls, issuing line load controls, and playing recorded announcements.

Recently the national phone network of AT&T altered its structure considerably. Instead of being structured hierarchically as explained above, a new approach called Dynamic Nonhierarchical Routing (DNHR) is being used [Ash and Mummert 1984]. The system is termed dynamic because a call may be routed over different paths at different times of the day to take advantage of spare network capacity. The system is termed nonhierarchical because switches are no longer separated into a hierarchy of different classes; they are equivalent in function. In short, any call may be routed through any part of the network to reach its destination.

As Mocenigo and Tow [1984] point out, managing the DNHR network is analogous to finding "a moving needle in a moving haystack" due to the increased dynamic nature of the system. Recent research efforts at AT&T have been directed towards introducing a higher degree of intelligence into the system, thus automating the system to an even greater degree. As Mocenigo and Tow note, however, it will not be possible to eliminate the role that the human monitor must play in this system, largely due to the problems that system failures create.

A very different type of large-scale system, the Army's Communications-Electronics Management System (C-EMS) [U.S. Army 1977a, 1977b], is designed as a means to meet the communications requirements of the battlefield. Due to

the dynamic nature of its environment, the system lacks the permanence and the level of automation of the nationwide phone system. In addition, since the military must be mobile during combat operations, a high degree of engineering and planning is required to produce an integrated, effective system. During a battle, for example, parts of the network may be damaged or communication units may change locations. The system must be able to respond quickly to these changing resource capacities and network configurations.

The C-EMS is composed physically of several different forms of communication device. Although the telephone network described above is composed largely of phone-related equipment, the C-EMS network may consist of radio, wire and cable systems, radio-wire integrated systems, messenger services, and visual and sound communications. These system components typically are arranged hierarchically in a manner similar to that of the phone network. Each device is referred to as a node. The specific system requirements are dependent upon the type of information to be transmitted, the form in which it will be received, and the security and speed required. Thus, not only is the C-EMS more mobile and less automated than the nationwide phone system, but it can also be described as less homogeneous.

## 2. MABEL

MABEL (Monitoring, Accessing, Browsing, and Evaluating Limits) [Henneman and Rouse 1984b; Henneman 1985a] is programmed in Pascal on a VAX 11/780 computer and operates in real time. It is structured as a large network that can range in size from hundreds to thousands of nodes. Customers travel through the system from a randomly selected source node to a random destination. Subjects monitor this system activity via a CRT display. When they detect a problem in the system (possibly due to a failure), subjects issue an

appropriate command through a keyboard to correct and compensate for the abnormal situation. The overall system objectives are: 1) to maximize the number of customers served, and 2) to minimize the time it takes for customers to travel between their source and destination nodes.

The following sections discuss MABEL in more detail. Emphasis is placed on the structure of MABEL, the operator interface, and typical system operation.

System structure. Several elements are basic to the structure of MABEL. A node represents the smallest structural unit in the network. Customers are processed at nodes with service times following an exponential distribution. Each customer is passed from node to node, following a path that will minimize its expected time in the system. If a node in a customer's path is currently busy, the customer joins a waiting line at that node until the node becomes idle.

As mentioned above, MABEL can consist of hundreds or thousands of nodes. It is impossible for the human operator to perceive and process information about all of the nodes at one time. On a more practical level, it is impossible to display information about all of these nodes at one time. Thus, nodes are grouped into relatively small networks called clusters. Human operators are restricted to viewing only one cluster at a time on the MABEL display.

These clusters are grouped into hierarchical levels. A customer typically enters the system through a cluster in the lowest level. The customer proceeds through that cluster to a node that connects to a cluster in the next higher level. This process repeats until the customer reaches the top level of the system. At this point, the process reverses: the customer travels through a cluster and then "jumps" down to the next lower level. The process



repeats until the customer reaches his destination.

Thus, as noted above, the system is analogous to a telephone communications system. Imagine a hypothetical three level network in which a call is placed from Americus, GA to Mason City, IA. The message first travels from Americus to Macon to Atlanta via a network of telephone lines and switching stations. The message then travels from Atlanta to Chicago. It is then transferred to Davenport, IA, and finally proceeds to Mason City. Atlanta and Chicago are at the highest level of the hierarchical system; Macon and Davenport are at the second level, while Americus and Mason City are at the lowest level.

Customers initially arrive at a node in the lowest level in the system. These arrivals are scheduled according to a Poisson process. Routing through the system is completed automatically as determined by a shortest path algorithm. Thus, customers are routed through nodes that will minimize the time they spend in the system.

Operator interface. Subjects obtain information about MABEL from a video display (Figure 1). The screen is divided into several sections. The upper right portion of the screen displays a cluster of nodes. The dim numbers to the left of each node identify the node, while the numbers inside each node represent the current queue size (the number of customers waiting to be served). This portion of the screen is updated approximately every two seconds. A different cluster of nodes is viewed by entering an appropriate command.

The lower right portion of the screen is an aid to the user to identify the current displayed cluster. Each letter (A,B,C) represents a level in the hierarchy. Each number (1,2,3,...) represents either a node or a cluster.

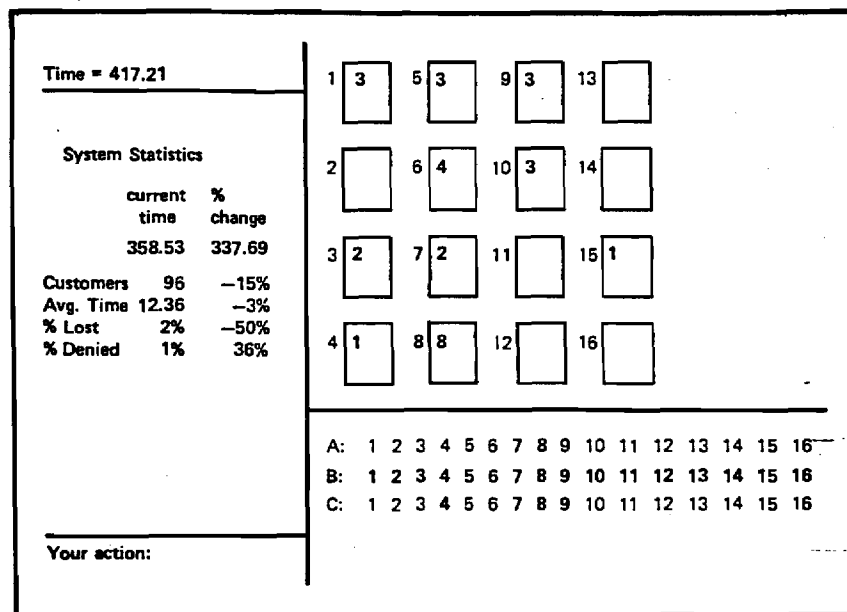


Figure 1. MABEL Display

Bright and dim characters are used to indicate the subject's current position in the hierarchy. A row of characters that is completely bright represents the cluster that is currently displayed on the screen. One bright character in a row of characters indicates the node above the currently displayed cluster. In Figure 1, therefore, the displayed cluster is in Level B. This cluster is beneath Node 7 of Level A.

The upper left portion of the screen is used to display the current time. Time is updated approximately every three seconds. Since the system operates in real time, customers will keep arriving to the system whether any action is taken by the operator or not.

The middle left portion is used to display a variety of user-requested information about the system. This information is input at the prompt "Your action:", located at the lower left part of the screen. Ten different commands are available to the user. These can be grouped into four categories: access, monitor, diagnosis, and control. The ten different commands are summarized in Table 1.

Typical system operation. Under normal circumstances, MABEL will operate automatically without any interference from the human monitor. When a node failure occurs, however, the human must act to diagnose and repair it. Node failures can occur in two ways:

- (1) Total failure due to malfunctioning equipment: in this case a node is unable to service any customers waiting at it. All customers waiting at

---

ACCESS Commands

d      down one level  
u      up one level

MONITOR Commands

m      monitor next level  
s      system summary statistics  
c      cluster summary statistics

DIAGNOSTIC Commands

t      test displayed cluster  
         for failures  
n      node information

CONTROL Commands

r      replace node  
l      reduce load

Table 1. MABEL Command Set

---

the node are lost, thereby reducing the queue size to zero. Additionally, the node refuses to accept any customers from another node. These customers are retained at their previous node. Since they are unable to proceed, the situation may lead to the following type of failure.

- (2) Failure due to exceeding the capacity of the node: each node has a maximum number of customers that it can "store" at any one time -- that is, each node has a maximum queue size. If this queue size is exceeded, the node fails. Its behavior after this point is identical to equipment failures. The node is unable to accept customers and, thus, new customers are retained at their old node. Once a failure occurs, therefore, it is likely to lead to other failures. In the extreme case, if nothing is done to repair failed nodes, the entire system will fail.

It is, of course, also possible for this type of failure to be induced simply by trying to service too many customers, i.e., the system is trying to handle too much of the load. In this case, customers arrive at a node at a rate faster than the node can service them.

Subjects locate failures by monitoring critical system states and testing suspect nodes or clusters of nodes. If a failure is found, the subject dispatches a crew to repair the node. If the system becomes too crowded with customers, the subject can issue a command to reduce the number of customers admitted to the system.

### 3. CAIN

Certain features of MABEL were substantially changed to develop CAIN (Contextually Augmented Integrated Network) [Henneman and Rouse 1986; Henneman 1985a]; however, the underlying structure of CAIN is identical to that of

MABEL. CAIN, however, is contextually augmented. The simulation has a much higher level of fidelity in that the addition of context produces a simulation with a much stronger resemblance to a real system.

Thus, although the physical hierarchical structure of MABEL was preserved, the addition of contextual information to CAIN required changing some features of the interface. In the MABEL scenario, for example, all nodes on a display page are identified by a number on the CRT display. Each displayed node in a cluster, therefore, is physically identical to nodes in other clusters. The MABEL interface has a generic quality in that all subsystems are visually similar; no contextual cues exist. On the other hand, nodes in CAIN are identified via specific geographic locations. Thus a node in MABEL with the label "9" might be labeled "Chicago" in CAIN. A typical CAIN display is shown in Figure 2.

Simply introducing geographic names as node labels is not enough, however, to alter subject task performance. A small experiment ( $n=3$ ) replicated the first MABEL experiment, with the exception that nodes were given geographic names. Subjects still referred to nodes by number only; contextual labels were present but not needed to perform the task. No significant difference was found in terms of performance between subjects using the two task scenarios. This result suggests that the addition of context must be such that it provides associative links (i.e., memory aids) or cues (i.e., clues to the location of problems within the system) through which subject performance is enhanced or task difficulty is decreased.

Associative links. The formation of associative links in CAIN is facilitated by the way in which a subject identifies a node. In CAIN nodes are referred to by geographic labels only, never by number. Subjects may input

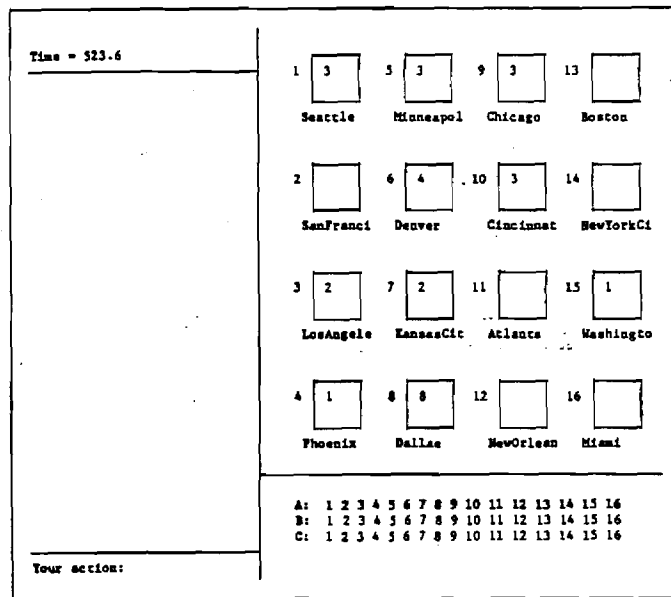


Figure 2. CAIN Display

the shortest string of characters that uniquely identifies the node from all other nodes in the system. Thus, "Denver" may be abbreviated "den". Most nodes can be identified with a three or four character substring of the complete name. In addition, the number of elements on a display page is kept constant at 16 so that the contextual information is invariant.

To illustrate the effect this change has on the subject's task, consider the command that displays a lower level cluster. In MABEL, the subject inputs the command "d2," which displays the cluster beneath node 2. In CAIN, on the other hand, the subject types "dSanf," which displays the cities beneath San Francisco (e.g., Berkeley, San Jose). Thus, subjects can form associations or links between system parts due to the existence of contextual information.

Subjects can use these learned associative links to maneuver through the CAIN display hierarchy. In MABEL movement between display pages is constrained to the cluster of nodes immediately above or below the current display. Thus, it is not possible to jump laterally across the network. In CAIN, however, it is possible to move from one part of the system to any other part. For example, if a subject recalls that the cluster associated with Bangor, Maine, was previously experiencing problems, it is simple to call up that particular cluster display. In addition, subjects can always return immediately to the highest level in the system.

Cues. The formation of cues in CAIN is provided by the introduction of context-dependent events. These events are of one of two types: recurring failures and nonuniform loading. Although equipment in nodes fails randomly, some equipment experiences a higher probability of failure. For example, a thunderstorm in Little Rock, Arkansas, may make equipment in that city susceptible to lightning damage. Similarly, given that incidents of vandalism are more likely to occur in Newark, New Jersey than Council Bluffs, Iowa, there is a greater chance of equipment damage in Newark. Therefore, equipment in certain cities exhibits a greater tendency to fail than in other cities. Subjects are informed of these locations via warning alarms that appear on the bottom of the display. Subjects can directly monitor activities within these trouble spots via a special "watch" command.

Besides recurring failures, another type of context-dependent event present in CAIN is nonuniform loading. At different times, some sections of the system may be prone to heavy loading. For example, certain times of day are busier in one part of the country than in others. Similarly, a major political or sports event in one section of the country may increase the number

of messages sent. As with the recurring failures, subjects are told the location of these increased loads via a message at the bottom of the screen. Subjects can then reduce the number of customers admitted to the overloaded subsystem.

In summary, despite the structural isomorphism of the two simulations, CAIN represents a significant departure from the context-free scenario of MABEL. Through the addition of contextual detail and the addition of events that are dependent upon this contextual information, the simulation fidelity has been increased significantly.

#### 4. Issues

Initial efforts in this research program were to gain an understanding of the relationship between physical characteristics of a system and human performance in monitoring and controlling such a system. For example, information displays for computer-based large-scale systems are frequently constrained by their size: only a limited amount of information may be displayed at one time. Thus, the number of elements of a system presented at one time may affect the ability of the operator to perceive relevant system state information both rapidly and accurately.

Another system characteristic that may affect human monitoring and control performance is the number of hierarchic levels. A system with multiple levels may have a very strong effect, for example, on the length of time needed for a human operator to find a failed component. Although some guidance exists within the literature relative to trade-offs between depth and breadth in static display menu hierarchies [Paap and Roske-Hofstrand 1986], little guidance exists for dynamic systems. Finally, another system charac-



teristic of interest is the rate at which system components fail. If the main role of the human operator in a large dynamic system is to diagnose failures, an important issue is whether or not humans can change their control strategies to adapt to changes in the quality (or reliability) of individual system components. These system characteristics (display size, number of levels, and component failure rate) were considered in Experiment One.

## B. Experiment One: Empirical Analysis

### 1. Method

Twelve volunteer subjects were initially exposed to MABEL by a set of written instructions [Henneman and Rouse 1984b]. These instructions contained a detailed explanation of the overall structure and normal operation of MABEL, a summary of the commands, and an explanation of the subject's role in operating MABEL during off-normal situations. Summary sheets of this information were also available. A quiz verified subjects' understanding of the effects of failures on system performance.

Training concluded with a special version of MABEL that allowed subjects to stop the execution of the program at any time during the experimental run. This training version of MABEL had the advantages of allowing subjects to become familiar with the commands and become aware of the effects of failures on both display features and system performance without being overwhelmed by the progressive effects of failures. If a situation became too complex, the subject could simply halt the dynamic system, solve the problem, and proceed. Subjects supervised two different training scenarios: a system with 16 nodes/cluster and 2 levels, and a system with 9 nodes/cluster and 3 levels.

The experiment had three independent variables: cluster size (i.e., number of nodes per display) and number of levels functioned as within-subject factors and failure rate served as a between-subjects factor. Cluster size varied between 4, 9, and 16 nodes; number of levels varied between 2 and 3. Failure rate was defined as the probability that a randomly selected node in the system would fail during each iteration of the MABEL program. One iteration occurred after each activity in the network (for example, the arrival of a new customer to the system). Failure rate was either low (probability of failure/iteration = .0005) or high (probability of failure/iteration = .0010). The six subjects in each group controlled six systems corresponding to all possible combinations of cluster size and number of levels. The order of presentation to subjects was balanced in order to average out any residual training effect.

Performance was assessed in several ways. The measures can be broadly grouped into two categories, namely, product and process [Henneman and Rouse 1984a]. Product measures assess the final result of a problem solving session (such as number of customers served) and, thus, assess system-human performance. Process measures, on the other hand, assess how that result was obtained by evaluating individual steps in a subject's strategic approach to supervising the system.

The product measures calculated the length of time customers spend in the system (mean sojourn time) and the number of customers served during an experimental run. (These measures were normalized to account for inherent differences that exist among the different experimental system configurations. Henneman and Rouse [1984b] provide a description of this bias-correcting procedure.) Process measures were classified into three types: 1) errors (e.g.,

number of times a subject viewed a failed node but did not repair it), 2) failure diagnosis (mean time to diagnose a failure and the fraction of failures found), and 3) strategy (e.g., mean amount of time spent accessing, monitoring, diagnosing or controlling).

## 2. Results

Analyses of variance were performed to determine the effect of the independent variables (cluster size, number of levels, and failure rate) on each of the dependent measures. Overall trends within the product measures of performance were very consistent: performance degraded with increasing number of levels and improved with increasing display size. The effect of number of levels was very strong, producing up to a 5-fold degradation in level of performance. This effect was expected: the greater the percentage of nodes hidden from view, the greater the difficulty subjects experienced in supervising the system. For instance, the three-level systems resulted in substantially longer times to diagnose failures. Since it took more time for the effects of lower level failures to become obvious at the higher levels, the effects tended to be more serious than in the two-level systems. This lengthened diagnosis time tended to degrade most other performance dimensions.

A trend not predicted was that increasing cluster size would lead to improved performance. One would suspect that larger numbers of nodes per display should lead to increased task complexity. Thus, as the number of components that the human must deal with increases, performance should degrade. This is not the case with MABEL. A main reason for this result is that the larger systems are inherently more reliable than the smaller systems: the small systems contain fewer alternate paths between nodes through which customers can be rerouted following a failure. Thus, customers tend to be retained

more frequently at nodes when they have fewer alternate paths through the system. This system characteristic also accounts for the shorter failure diagnosis times found in the small cluster size systems: failures and their symptoms propagate faster in the small systems.

Failure rate did not play a role in shaping performance except with respect to the measures of strategy, e.g., the percent of time subjects spent performing different activities (accessing, monitoring, diagnosing, and controlling). Results suggested the prevalence of two basic strategies for supervisory control of MABEL. One strategy involved staying at higher levels and using monitor commands to assess the state of lower levels. The other strategy involved actually accessing the lower levels and performing tests to diagnose failures. Subjects with low failure rate conditions tended to select the former strategy, while subjects with high failure rate conditions tend to select the latter strategy. Apparently both strategies were effective in that performance was independent of failure rate. Thus, it appears that subjects could adopt strategies to compensate for decreased reliability of individual system components, but not for the more resource-constrained networks.

### C. Experiment Two: Measuring Complexity

The initial experiment considered the relationship between several physical characteristics of a large-scale system and human performance. A second experiment addressed this relationship more quantitatively by evaluating several measures of task complexity. Based on a review of the literature [Henneman and Rouse 1986], two measures of complexity relevant to the task of human monitoring and control a large-scale systems were proposed. Two dominant perspectives were identified within the complexity literature as being particularly relevant to this discussion, namely, that of the systems

scientist and that of the behavioral scientist.

Most studies of complexity performed by systems scientists are on a context-free or theoretical level. Although much work has gone into defining and measuring system complexity, little has been done to assess the implications of complexity. Furthermore, while humans must play an important role in many large-scale systems (e.g., failure diagnosis and network management), little research has investigated the relationship between large-scale system complexity and human performance. Finally, due to the strong theoretical flavor of the systems science approach, it is often difficult to see its application to real-world systems.

On the other hand, studies of complexity performed by behavioral scientists are on a very applied level. Although the approach often lacks the mathematical rigor of the systems approach, complexity is always related to some aspect of human performance. Unfortunately, differences between tasks and complexity measures cause difficulty in generalizing results across contexts. Moreover, the small, well-defined nature of the tasks seems to have little relation to human performance in large-scale system.

The research described in the remainder of this section attempts to integrate several perspectives concerning the nature of complexity, as well as illustrate the impact of this conceptualization of complexity on human performance in CAIN. Complexity is viewed as being a result of both the structure of the system and the human operator's understanding of the system. Complexity is also considered in terms of its relation to both system performance and human performance. Thus, both nonbehavioral and behavioral approaches are taken into account.

In this report, the complexity of a large-scale system is described in terms of: 1) the physical structure of the system and 2) operators' understanding of the system as reflected by their strategy. From this perspective, a system that is complex or difficult to control for one operator may be relatively easy to control for another operator. Similarly, the complexity of a system may vary with time for any particular operator. Some systems, however, may be complex regardless of any particular control strategy due to their inherent structural complexity. The following paragraphs propose two measures of complexity that incorporate these ideas. Structural complexity is considered first, followed by strategic complexity.

#### 1. Complexity measures

Structural complexity. A one-to-one relationship exists between the simulated physical structure of CAIN and the actual structure of the display-page hierarchy. Since the main control task in CAIN is to locate failures, a measure of structural complexity should assess the difficulty of finding failures given the physical arrangement of the system. A major constraint placed on an operator's ability to locate failures is the hierarchical display structure; thus, it seems reasonable that structural complexity can be estimated by calculating the total number of display pages the operator must view in order to repair all system failures. Assuming that the operator knows the location of all failures, this measure represents the minimum number of pages necessary to find all system failures. Therefore, the structural complexity measure represents optimal performance given the constraints of the structure or arrangement of the system components. Operator performance affects this measure only in that individual operators may have more or fewer failures depending upon their fault finding ability.

To illustrate how this measure is calculated, consider the system in Figure 3. This hypothetical system contains four nodes per display page and has three levels. Each group of four rectangles represents a cluster of nodes (i.e., one display page). For clarity, only those clusters of nodes that enter into the complexity calculation are shown. The darkened rectangles represent nodes that have failed. In this example, therefore, three failures exist within the system: two on the second level and one on the third level.

The structural complexity measure is determined by counting the number of display pages that must be viewed in order to find all failures. The counting method assumes a strategy based on tracing higher level symptoms to their lower level causes. (Context-specific cues might, of course, allow operators

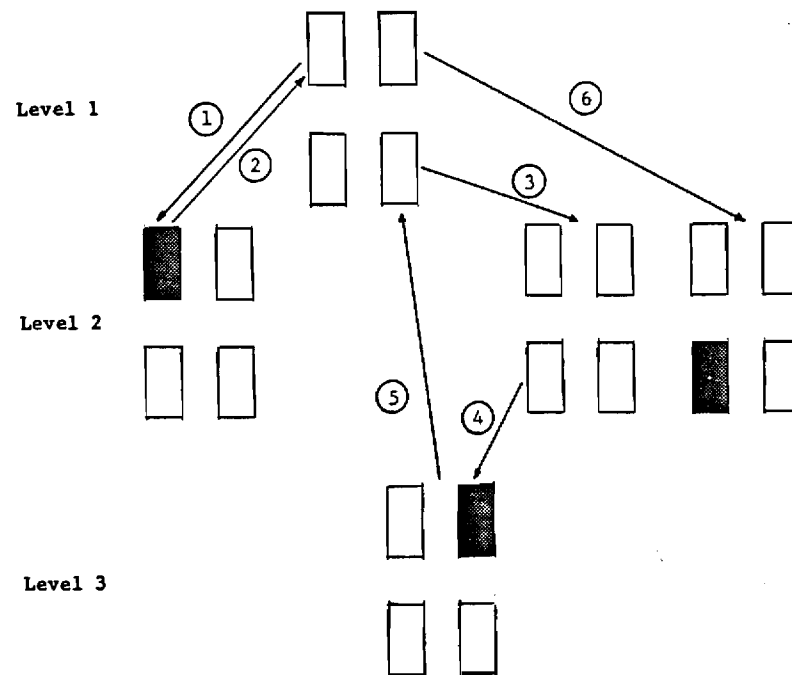


Figure 3. Example calculation of structural complexity.

to locate failures in fewer pages.) Thus, the counting method assumes that after locating all failures along one subsystem branch, the subject returns to the highest system level to search the next branch (a depth-first strategy). Figure 3 is self-explanatory: to repair all three failures in the system, an operator must view at least six display pages. The final return to the top system level is not counted into the measure because it would simply add one to all estimates.

Strategic complexity. The strategic complexity measure explicitly considers operator performance. When operators are deciding which path through the system is most likely to lead to finding a failure, they make a tradeoff between their uncertainty concerning the state of a subsystem display page (i.e., queue lengths) and their expectations of finding a failure in that subsystem. High uncertainty about a subsystem may be acceptable, for example, if a relatively low probability exists of finding a failure on that display page. On the other hand, high subsystem uncertainty may be unacceptable if a very high probability exists of finding a failure.<sup>1</sup>

State uncertainty  $U$  is defined as the real time elapsed since a particular display page was last tested for failures. Probability of failure is defined as the probability that a failure exists within a cluster, given the state of the display  $X$ , and is denoted by  $p[F|X]$ . For example, when a subject views a particular display page, features of that display provide information about the existence of failures in other subsystems (e.g., a large queue size

---

<sup>1</sup>Of course, the acceptance of uncertainty will also vary as a function of the consequences of a failure. If, for example, a failure is likely to lead soon to another failure, high uncertainty about that subsystem would be unacceptable. Subjects, however, do not have any knowledge of these possibly unequal probabilities. Thus, it is reasonable to assume equal effects of failures for this discussion.



suggests a lower level failure.) Experimental data files were replayed in order to estimate these probabilities empirically. These probabilities were determined by dividing the frequency with which a display state reflected a failure by the frequency with which a particular display state (i.e., queue length) was viewed by an operator. Sets of probabilities were calculated for different system configurations (2 vs. 3 levels and high vs. low redundancy) and different loading rates (e.g., a system with a low loading rate has fewer customers in service and, hence, lower threshold or queue size will reflect failures).

The measure of strategic complexity multiplies these two measures (state uncertainty and probability of failure given the system state) and sums the product across all clusters in the system:

$$\text{strategic complexity} = \sum U(i) * p[F|X(i)]$$

where  $U(i)$  is the time since last accessing display page  $i$ ;  $X(i)$  is the state of page  $i$  reflected by the display one level higher;  $p[F|X(i)]$  is the probability of failure given state  $i$ ; and  $F$  denotes "failure." Strictly speaking, this conceptualization results in strategic complexity having units of seconds. The  $U(i)$  values are really just "proxy" measures [Keeney and Raiffa 1976] of complexity, however, and thus, strategic complexity is left unitless.

When a subject descends to a lower level, the  $p[F|X(i)]$  remain fixed for the previous level. When a subject returns to the higher level, the  $p[F|X(i)]$  value associated with the just-visited lower-level clusters is set to zero. Thus, when an operator descends to a lower-level subsystem and tests for failures, the strategic complexity measure is simultaneously increased by the "new" uncertainty (i.e., increased  $U(i)$ ) present in the other lower-level subsystems and decreased by the certainty (i.e.,  $U(i) = 0$ ) now associated with

the current level.

To illustrate how the strategic complexity measure is determined, consider the display in Figure 4. This system contains four nodes per display page and has two levels. The operator is viewing the highest level page in the display hierarchy and is monitoring activity in the next level of the system. The operator can gather information about activity in the second level of the system from two sources in this example: the cluster display and the data displayed via the monitor command. The monitor command lists the number of customers in the clusters one level below; the cluster display shows the number of customers waiting at all nodes in the current cluster.

Each of these pieces of information reflects the probability that a failure has occurred in a lower level cluster. These probabilities (which are plausible but hypothetical) are listed in Table 2. For example, the queue size of 15 in Denver reflects a relatively high probability (0.75) that a

---

<u>Monitor Display</u>		<u>Cluster Display</u>	
<u>Cluster</u>	<u>Number of customers</u>		
Denver	8	<div>15</div>	<div>1</div>
Los Angeles	5	Denver	Chicago
Chicago	1	<div>2</div>	<div>5</div>
New York	1	Los Angeles	New York

Figure 4. Example monitor and cluster display for calculation of strategic complexity.

---

---

<u>Cluster</u>	<u>U</u>	<u>p[F X]</u>		<u>U x p[F X]</u>
	<u>uncertainty</u>	<u>monitor</u>	<u>cluster</u>	
Denver	20.12	.600	.750	15.090
Los Angeles	0.54	.100	.015	.054
Chicago	7.36	.001	.001	.007
New York	9.12	.001	.050	.456
Strategic Complexity =				15.607

Table 2. Example Calculation of Strategic Complexity

---

failure exists in level two. Similarly, the monitor command reports that eight customers are currently in the cluster beneath Denver; these eight customers reflect a 0.60 probability that a failure exists. The operator has not tested the cluster beneath Denver for failures for  $U(\text{Denver}) = 20.12$  s. Using the information that reflects the highest probability of failure (i.e., from the cluster display) results in the following measure of strategic complexity for the Denver region:

$$\begin{aligned}
 U(\text{Denver}) * p[F|x(\text{Denver})] &= 20.12 * 0.75 \\
 &= 15.09
 \end{aligned}$$

This procedure is then repeated for the other clusters in the network and the measures are added together. In this way the total strategic complexity is determined to be 15.61.

In this example, it should be noted that Denver makes a large contribution to the strategic complexity measure for two reasons: first, the operator has a high degree of uncertainty concerning the Denver subsystem in that it

has not been tested for failures in 20.12 s. Second, the display reflects a very high probability (0.75) that a failure exists in the Denver subsystem. The combination of these two factors leads to a very high measure of strategic complexity for the Denver subsystem. On the other hand, the other subsystems have either a low uncertainty measure or a low probability of failure. Thus, as shown in Table 2, their contribution to strategic complexity is small.

Dependent measure of complexity. The literature review also suggested that an appropriate dependent measure of complexity is the time to failure diagnosis. In the context of CAIN, this measure is the mean time from when a failure occurs to when the subject issues a repair command for a failed node. Since the two independent complexity measures vary with time and since there are multiple repairs occurring in conjunction with the assessment of the variables, it was necessary to use a dependent measure that also changes with time. Average time, therefore, includes the diagnosis time for the current repair plus diagnosis times for the four previous repairs.

To summarize, the structural measure reflects an inherent characteristic of the network, namely, the number of display pages necessary to find all the failures in the system. The strategic measure, on the other hand, reflects temporal aspects of subjects' strategies, i.e., subjects' paths through the network. From this perspective, the strategic measure reflects the complexity resulting from a particular strategy.

Although the two complexity measures proposed here may have some general applicability (in particular, the measure of strategic complexity is appealing due to its temporal nature), it is not the intent of this work to suggest or prove that these measure are true indices of task complexity. The goal instead is to show in a pragmatic sense that these two dimensions represent a

useful distinction relative to task complexity. These measures are a convenient means to demonstrate this distinction.

## 2. Method

The main goal of Experiment Two was to investigate the nature of complexity in a large-scale human-machine system. As emphasized in the preceding section, the general assumption is made that task complexity can only be measured relative to an individual's understanding of the system and expertise in dealing with problems in that system. Thus, complexity is considered to be dynamic, varying across time and among subjects. Accordingly, as discussed below, subjects were required to perform the task (CAIN) over a relatively long period of time.

Results from the Experiment One indicated that cluster size (number of nodes per display page) in MABEL had a particularly strong effect on subject task performance. Results suggested that small clusters degraded performance because fewer connections existed between nodes; less redundancy caused failures to propagate more quickly. Another result from Experiment One showed the very strong effect of number of hierarchical system levels on human performance. Thus, two independent variables selected for further analysis were the degree of redundancy (or connectivity among components) and the number of levels in the system. (Cluster size was kept constant at 16 in order to emphasize the nonvarying features of the contextual display.) Redundancy or connectivity was defined as the number of connections emanating from each node. Redundancy varied between low (six connections between nodes) and high (13 connections per node), and the number of levels varied between two and three.

Of interest in this experiment was the way in which complexity changes as subjects gain expertise. Thus, the order of presentation of experimental conditions was not randomized. All subjects saw the same experimental conditions in the same order. A final independent variable, therefore, was the order of presentation of experimental conditions within each combination of number of levels and redundancy.

Eight paid subjects were trained in three sessions via a combination of written instructions and hands-on experience with CAIN, similar to that used in Experiment One. Subjects completed the first two training sessions by controlling a two-level CAIN system. The third training session was spent controlling a three-level CAIN system. As in Experiment One, these sessions were performed using a version of CAIN that allowed subjects to start and stop the program execution.

Summarizing the ten experimental sessions (S1-S10), they were performed in the following order (with the intent of increasing experimental difficulty): S1 and S2 had two levels with high redundancy; S3, S4, and S5 had three levels with high redundancy; S6 and S7 had two levels with low redundancy; and S8, S9, and S10 had three levels with low redundancy. Each experimental session was performed on consecutive days and lasted about 45 minutes.

### 3. Results

Summary of Approach. Data from this experiment were first analyzed using the same performance measures used in Experiment One, e.g., mean time to failure diagnosis and fraction of failures repaired. Overall results from the analysis of variance of subject performance measures supported those of Experiment One. Measures of fault diagnosis performance were affected as expected

by the independent variables. Increasing the number of system levels from two to three corresponded to a higher mean time to failure diagnosis. This result was largely because failures take longer to propagate upwards in the three-level systems. In addition, failure-related symptoms take longer to emerge in highly interconnected networks; thus, the high redundancy systems resulted in longer mean times to diagnosis. The fraction of failures repaired by subjects was also significantly affected by increasing the number of levels: as the number of levels increased from two to three, the fraction of failures found decreased from 0.95 to 0.69. As in Experiment One, subjects had difficulty coping with the very large search space in the three-level systems.

The data were also analyzed with the purpose of investigating relationships between the complexity measures, the CAIN environment, and operator performance. This investigation was accomplished in two ways. First, an analysis was undertaken of average or global measures of complexity (i.e., the complexity time series averaged over each experimental run). The effect of the experimental independent variables (number of levels and degree of interconnectivity between nodes) on the mean complexity measures was determined using analysis of variance. The relationship between the mean complexity measures and measures of subject failure-diagnosis performance was then assessed by using correlation analysis. As is discussed below, this analysis of mean complexity values provided explanations for differences that exist between different system configurations.

The second way in which complexity was investigated involved using a fine-grained approach, namely, time-series analysis. Time-series analysis was selected due to the intrinsic time-varying nature of the independent and dependent complexity measures. This analysis provided insight into the way in

which complexity evolves and affects different phases of the failure-diagnosis process.

Due to the amount of time necessary to perform these analyses, the results are limited to Sessions 2, 5, 7, and 10. Data for the analyses were generated by replaying subject data files. Every three seconds (corresponding to the rate of display update), both complexity measures and the mean time to failure diagnosis were calculated. Mean values for all measures were calculated from these time series.

Analysis of average complexity measures. The results of two ANOVAs using mean structural and strategic complexity measures as dependent measures and number of levels and degree of redundancy as independent measures are summarized in this section. Structural complexity, as measured here, was decreased in two ways: 1) decreasing the number of system levels and 2) decreasing the number of system failures. The first way enables subjects to access fewer display pages in order to diagnose failures in the lowest system level. The second way is facilitated by increasing the network redundancy (i.e., increasing the number of connections between nodes). As network redundancy increases, the mean number of node capacity failures decreases, which has the effect of decreasing the structural complexity measure.

Strategic complexity, as measured here, may be decreased in three ways: 1) using an effective strategy in terms of responding to symptoms, 2) decreasing redundancy, and 3) decreasing number of levels (which causes symptoms to emerge more rapidly). Subjects tended to trace failures to the lowest system level only when a symptom (i.e., visual cue) appeared on the display, even if they had not viewed a particular region in a large period of time. Consequently, when symptoms emerged slowly (as in the high-redundancy/three-level



conditions), high uncertainty resulted. This uncertainty helped to create moderate to high strategic complexity. On the other hand, symptoms emerged more rapidly in the low-redundancy/two-level conditions. Since operators tended to wait for symptoms to emerge on the top-level display, low redundancy led to low values of strategic complexity.

This dependence on visual cues has implications for the design of task performance aids. One possibility is to have aids that help people to overcome their inability or reluctance to reduce system uncertainty despite the absence of failure symptoms. Alternatively, failure-related cues or symptoms could be enhanced so that operators naturally pursue leads sooner.

These results provide insight to the overall characteristics of the two complexity measures and their relationship to subject fault diagnosis performance. The measures are sensitive to variations among the system characteristics of number of levels and degree of redundancy. In general, the more complex systems have three rather than two levels. Although multiple system levels might be desirable in that they allow supervision of large networks and protect upper levels from the effects of failures, they have the undesirable side effect of masking symptoms from operators, thereby increasing the complexity of failure diagnosis. Multiple displays could possibly be used to reduce this complexity. The effect of redundancy on complexity depends on the type of complexity: more redundant systems (corresponding to less structural complexity) enhance the proper operation of the system by reducing the impact of failed components. On the other hand, more redundancy leads to increased strategic complexity (the complexity of failure diagnosis) due to the slower emergence of failure symptoms.

Beyond the characteristics of these single complexity dimensions is another important conceptual and methodological issue: the multidimensional nature of complexity, i.e., the relationship between the independent and dependent measures of complexity. A correlation analysis between the two average complexity measures and the two independent measures indicated that when many failures exist in a system, the general tendency is for the complexity measures to increase. At the same time, however, the mean time to failure diagnosis decreases. Thus, even though complexity may be large, failure-diagnosis time may be small.

This observation emphasizes the distinction mentioned previously between proper system functioning and the complexity of failure diagnosis. In a localized sense, control in a complex system is simple: no matter what the operator does, it will result in finding a problem (as reflected by short diagnosis times). In a global sense, however, control in a complex system is complex: so many problems may exist in the system that proper operation is endangered, as reflected by a low fraction of failures found. The operator, dealing with only a small part of the system at one time, may be oblivious to the scope of problems in the network. Another important issue is, therefore, the impact of a richly interconnected multiple-level system (that supports proper system functioning) on the complexity of human monitoring and control (that will degrade failure diagnosis performance).

Analysis of fine-grained complexity measures Time-series analysis [Box and Jenkins 1976] was used to identify, estimate, and diagnostically check transfer functions that relate the two input complexity measures (structural and strategic) to the mean time to failure diagnosis. Each transfer function model predicted the current mean time to failure diagnosis through a linear

combination of the complexity measures at various time lags. The essence of the modeling process was to determine the time lags to include in the model and the weight or relative contribution of each time-lagged variable to the predicted value.

Overall, the approach was successful. The equations removed all structure from the autocorrelation function of the model residuals. Furthermore, a comparison of the sum of squares of the original dependent time series (i.e., mean time to failure diagnosis) to the sum of squares of the residuals showed that the transfer functions explained 82 to 97 percent of the variance within the original data. Nevertheless, wide differences in the lag and coefficient values in the models existed among both subjects and systems.

A plausible explanation for these differences was derived by identifying certain characteristics of the task, the system, and the human related to the the process of failure diagnosis. For example, several different events are associated with the life cycle of each system failure: failure occurrence, symptom emergence, and failure diagnosis. Failure occurrence is when a part of the system fails. Symptom emergence is the time period between failure occurrence and the time a failure first affects any node that appears on the subject's video display. Failure diagnosis is the time period from failure occurrence to when a subject issues a repair command for a failed component. The timing of these events undoubtedly has some effect on the length of time needed to find the failure. Moreover, the system complexity at these event times might also affect failure diagnosis time.

Besides the possibility that different events associated with the failure life-cycle affect diagnosis time, it is also reasonable that different types of diagnosis might affect failure diagnosis time. The diagnosis of any

particular failure may be classified as one of three types: topographic, symptomatic, or serendipitous. Subjects identifying failures using a topographic strategy trace failure symptoms from higher system levels to their causes in lower levels. Subjects identifying failures using a symptomatic strategy make a direct mapping from their knowledge of the system structure to the failed component. A symptomatic diagnosis relies, therefore, on the subject's contextual knowledge of the system. For example, when subjects make a jump from one cluster to another in the same level to repair a failure, their action suggests that their context-specific knowledge of the system is providing guidance to system trouble areas. Finally, subjects may also identify failures accidentally or serendipitously. In this diagnosis mode, subjects locate failures while browsing through the system or while tracing the cause of a different failure.

In summary, it is possible that several different types of failure-related events (e.g., failure occurrence and symptom emergence) and several different modes of failure diagnosis (e.g., symptomatic, topographic, and serendipitous) affected the time to failure diagnosis within a system. In addition, due to the aforementioned averaging window of five failure-diagnosis times for the dependent complexity measure, it is possible for many lags (possibly quite long) to have entered the transfer function. From the perspective offered in the preceding paragraphs, therefore, the transfer functions relating the two complexity measure to failure diagnosis time were affected by types of failure-related event, modes of failure diagnosis, and the way in which diagnosis time were aggregated.

These factors were considered analytically by replaying subject data files and comparing measures of the characteristics described above to the

transfer functions. Results showed that the variables and lags present in the transfer function were reasonable, if not entirely explainable. The real-time values of lags frequently agreed with the mean inter-failure event times calculated from subject data files. A comparison of these values suggested that recurring patterns of agreement existed between the lags and inter-event times. These recurring patterns were useful to explain the presence of both positive and negative terms in the transfer functions. Differences between time values can probably be accounted for by any of several reasons, including the high variability present within the data, the subjective nature of the modeling process, and the existence of events other than failure occurrence or symptom emergence (e.g., diagnosis time for a particular system level or subsystem) that affected parameters in the transfer functions.

These results demonstrate how two different dimensions of complexity, structural and strategic, can be related to human fault-diagnosis skill in a large-scale system. The exact nature of the two measures is relatively unimportant beyond a certain degree of intuitive validity. The importance of these results, however, lies in the demonstration that the complexity measures were dependent upon the number of failures in the system and the rate at which their symptoms emerge. These factors were highly dependent upon both system characteristics (i.e., number of levels and degree of redundancy) and subject strategy. Of equal importance is the demonstration that the complexity measures related to performance in a time-varying manner, and the nature of this time-varying manner was highly dependent upon events that occurred within the system and the strategy of individual subjects.

#### D. Conclusions

The experiments, results, and conclusions up to this point have considered the relationship between the design of a large-scale system and human monitoring and control behavior. System characteristics such as number of levels and degree of interconnectedness can have a very strong effect on the ability of humans to maintain proper system operation in the presence of failures. Since the normal system operation tends to be affected in the opposite direction in the presence of the same design characteristics, system designers must be careful to create environments that support both system and human performance.

Straightforward measures were used to assess the complexity of a large scale system as it relates to the task of monitoring and control. Complexity, as discussed in this report, is a dynamic property of a human-machine system. Complexity varies with time and it varies among operators. Furthermore, complexity is multidimensional: two dimensions of complexity (i.e., structural and strategic) have been proposed, and it appears that this distinction is useful both conceptually and practically. Complexity is not due solely to the structure of the system, although a system may certainly be complex due to its structure. Rather, complexity also arises when the human, trying to solve problems within the system's environment, does not understand the structure and as a result issues an inappropriate command, misinterprets display information, etc. In short, systems are also complex due to humans' understanding of the system as reflected by their strategies.

Another result from this work concerns the outcome of complexity. Based on a review of the literature and the major control task of subjects (i.e., finding failures), mean time to failure diagnosis was used as the major

dependent measure of complexity. As results suggest, however, mean time to failure diagnosis alone does not completely describe the implications of complexity. For example, the most complex systems resulted in shorter failure-diagnosis times due to the number and location of failures. A smaller fraction of the total number of failures was diagnosed, however. Thus fraction of failures diagnosed was used to explain a different aspect of performance related to task complexity. In short, the result of complexity is multidimensional. A single dimension does not capture the outcome of a complex system.

These comments are important in light of the relationships among system characteristics that contribute to complexity, proper operation of the system, and complexity of monitoring and control by the human. As the system becomes more "complex" (from a nonbehavioral perspective, i.e., more levels and more redundancy), it becomes more resistant to the effects of system failures. Failures take longer to propagate through the more complex systems. Moreover, the effects of any one failure on overall system performance are minimized due to the number of alternate paths through the system. Hence, normal system operation is enhanced. This situation is analogous to the use of redundant or standby equipment in systems to increase fault tolerance. On the other hand, as the system becomes more complex, the task of finding system failures becomes more difficult. Although the system design characteristics can help to avoid the short-term effects of failures, they can have the dual effect of making the human supervisory controller's task more difficult. These findings lend support to Nawrocki's [1981] conjecture that efforts to simplify the task of equipment operation through hardware design tend to complicate the task of equipment maintenance.

The relationship between complexity and human performance takes on increasing importance given the growing prevalence of large-scale systems. Human abilities and limitations in monitoring and controlling these complex systems must be identified in order to design systems that facilitate good failure diagnosis and network management performance. In short, systems must be designed such that they do not overload human information processing capabilities. Beyond the issue of design, an understanding of human performance constraints should facilitate the creation of effective performance aids. Such aids can be used to help people overcome their limitations in coping with the complex environments these systems create, thereby leading to safe and effective system performance.



### III. MODELING HUMAN PERFORMANCE

The work described in the preceding sections has implicitly modeled human performance as a function of various system characteristics. Powerful statistical evidence illustrated the strong effect that a system designed for good automatic control can have on a human operator's ability to exercise accurate and timely system intervention. From a behavioral viewpoint, however, the statistical models that have been described do not offer sufficient cognitive explanation for human performance. The empirical analyses describe what happens when humans interact with a large-scale system, but they do not help to explain why things happen that way. Therefore, the second phase of this research program concentrated on the development of a behaviorally valid model of human performance in monitoring and controlling a large scale system [Henneman 1985b; Zinser 1986; Zinser and Henneman 1986].

Modeling is a good approach in this problem area for several reasons. First, a modeling approach will contribute to a better understanding of human performance in this task. From the previous experiments, much knowledge (both formal and anecdotal) was obtained about how people perform this task. The modeling process allows the formal codification of knowledge and cognitive mechanisms relevant to a complex monitoring and control task. Both human abilities and limitations must be identified by this process. Thus, the model should contain appropriate knowledge representations and implementation mechanisms to provide a high level of behavioral fidelity to human performance.

Second, the modeling approach should facilitate the development of an approach to aiding the human operator. A model that incorporates mechanisms coherent with human cognitive functions should be able to provide meaningful

and timely aid to the human operator. Thus, a focus of this work is the use of the model as the basis of an on-line human performance aid.

#### A. MURRAY : A Model Of Human Problem Solving

The model developed in this report is an extension of a conceptual model of human problem solving proposed by Rouse [1983]. Rouse has suggested that problem solving takes place on three levels: 1) recognition and classification, 2) planning, and 3) execution and monitoring. Thus, when a problem situation develops, the first task is to detect that the problem exists and to categorize it (recognition and classification). An approach or plan to solving the problem must then be developed (planning), and finally, the plan must be implemented (execution and monitoring). The model is further characterized by its ability to make either a state- or a structure-oriented response, depending on both the system state and the human's level of expertise. The model assumes that humans have a preference for pattern-recognition solutions to problems -- that is, humans prefer to make context-specific state-oriented responses to situations. Moreover, the model operates heterarchically at all three problem solving levels almost simultaneously, with situations constantly being re-evaluated relative to their state- or structure-oriented status.

Several efforts have used this generic problem solving model. Domains have included automotive and aircraft powerplants [Hunt and Rouse 1984], process control networks [Knaeuper and Rouse 1985], and communication networks [Viteri 1984]. Performance of these models was, in general, quite good; however, they were constrained by the lack of real "understanding" of the domain by the model. The models lacked knowledge structures that would allow flexible performance strategies to be pursued. Thus, results from efforts at using Rouse's model as the basis for an on-line performance aid [Knaeuper and Morris

---

1984] were equivocal. A major reason appeared to be the rigidity of the performance strategy of the model.

## 1. Overview of the Model

The model proposed for the CAIN environment, MURRAY, is illustrated in Figure 5. MURRAY operates in the three stages of Recognition and Classification, Planning, and Execution and Monitoring. Situations are continually re-evaluated as system states change due to the system dynamics or operator actions. An important feature of this task is that at any given time the human operator may have several different tasks that could be performed. The key to good performance is the ability to choose among these possibly conflicting subtasks. These model components, their associated representations, and how they interact will be considered below. The section concludes with an example of how the model operates.

## 2. Knowledge Representation

MURRAY's fidelity to human performance is dependent on the representation of three different types of knowledge needed to perform the task: system knowledge, contextual knowledge, and task knowledge. System knowledge and contextual knowledge are shown explicitly in Figure 3, while the task knowledge is embedded within the Recognition/Classification and Planning components. The Execution component of the model is realized by implementational procedures and the command that is issued.

The first type of knowledge, system knowledge, consists of information from CAIN about the current system state, e.g., the number of customers waiting to be served in a city. Thus, the system knowledge of MURRAY is identical to the information presented on the CAIN display. System knowledge is only

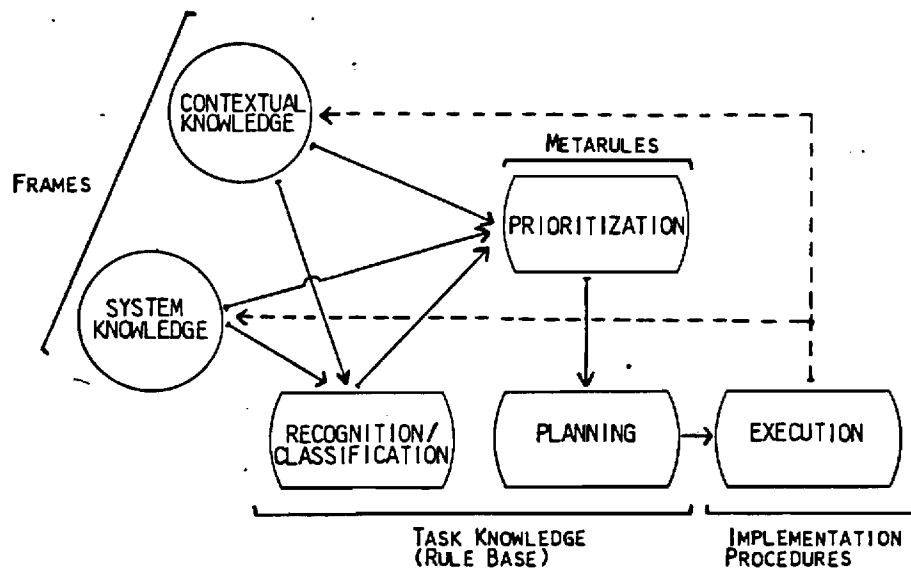


Figure 5. Components of the Conceptual Model

accessed by the model's Recognition/Classification component and the Prioritization mechanism. The system knowledge is structured as a hierarchical frame system [Minsky 1975]. The frame of the highest structural level represents the cluster currently displayed by CAIN. A cluster frame contains information regarding its location relative to other clusters and levels in the network. A cluster frame also contains 16 city frames that correspond to each of the cities (i.e. communication nodes) in the cluster. Each city frame has several "slots" that contain such information as the number of customers waiting for service at the city and the average length of time they have been waiting. These slots are either filled by data that appear on the CAIN display or by appropriate default values. The information contained within this set of frames will change as the information on the display changes. If a new

cluster is displayed, the slots in the 16 city frames change to reflect the features of the newly displayed cluster. The slots in the cluster frame will also inherit information from the city of the level above.

The second type of knowledge, contextual knowledge, consists of information concerning the context of the system at a given time, such as locations of individual cities in the network and cities that have high loading and abnormal failure rates. Thus, contextual knowledge is augmented over time; as the model gains 'expertise', the knowledge stored by this component will change. Contextual knowledge is represented by a network of context frames. This network contains a hierarchy of city and cluster frames as described above, and also data structures that describe both the evolution of the system to the current state, and the human operator's monitoring behavior and knowledge about the system at any given time. Since the human operator's knowledge of the contextual features of individual cities (e.g., high failure rate) and the contextual relationships among cities (e.g., Decatur is associated with Atlanta) will vary with time, the model's contextual knowledge also is augmented as time and, hence, experience increases.

Finally, the third type of knowledge, task knowledge, represents the operator's behavior in monitoring, problem solving, and failure detection. In other words, task knowledge refers to the knowledge needed by operators to perform their jobs, for example, repairing failed equipment. Task knowledge is represented as a production system [Newell and Simon 1972]. The operator's heuristics correspond to productions (or rules), while the operator's internal model of the system corresponds implicitly to 'metarules' that organize the application of the 'normal', explicit rules. The metarules are directly implemented in the procedures of the model's Prioritization component (or inference

engine), which will be explained later.

MURRAY contains 22 rules in its representation of task knowledge. These rules have a fixed syntax, and thus, they can be manipulated from outside the program by a text editor. The set of rules is based on a combination of expert judgement and empirical evidence from Experiments One and Two. Each rule consists of a situation and an action part made up of predicates. The situation part of a rule contains one or more predicates. Each predicate may have a value associated with it that relates to either the system or contextual knowledge of MURRAY. The predicates of a rule's condition part have the function of matching that rule to a CAIN system state or recalled contextual information. Thus, the condition part of a rule corresponds to the Recognition and Classification component of the conceptual model in Figure 5. The action part of a rule contains a command for CAIN. Thus, the identification of a set of potential actions corresponds to the Planning component in the conceptual model.

To summarize, the model depicted in Figure 5 consists of three interacting types of knowledge. System knowledge includes system state information as presented on the CAIN display screen. Contextual knowledge is also acquired from the display, although it is less transient in nature. Contextual knowledge is acquired over time and represents some of the long term relations among system components. Thus, both system and contextual knowledge can be thought of as forms of declarative knowledge [Anderson 1976]. Task knowledge, on the other hand, is a form of procedural knowledge [Anderson 1976] that delineates how the task should be performed. Details of the implementation of these representations in MURRAY and the way they interact are considered in the next section.

---

### 3. Implementation

An important part of MURRAY is the inference mechanism of the rule base representation of the task knowledge. This mechanism determines the way that rules are applied and evaluated. The mechanism is implemented whenever the system state changes, i.e., whenever the model observes a set of new data from CAIN (as a reaction to a command issued by the operator or a dynamic change in the system). At this point, the condition predicates of all the rules are evaluated successively in the Classification component of the model. Those rules whose condition parts match are then prioritized. This prioritization is partially based on a priori importance weights that are associated with each rule.

These a priori importance weights of applicable rules are dynamically altered by the characteristics of the current system state to which a rule is applied. Fuzzy set theory methodologies are applied to the set of all applicable rules. The use of fuzzy sets can be regarded as a means of representing the phenomenon of activation levels involved in human cognitive processes. (Hunt and Rouse [1984] describe a similar use of fuzzy sets in human performance modeling.) Three factors alter the initial importance weight values to determine the actual importance of a rule in a given situation. These factors are described below.

The first factor is determined by linear fuzzy membership functions. Several of the rules contain fuzzy predicates that describe the current system state in the form of qualitative expressions such as 'high' or 'low'. These values are used to define the membership of the rule in the set of applicable rules. The deviation of the value of the current system state from a 'normal' system state is proportionally weighted by the membership function. Thus, a

city with a queue size of 16 customers yields a higher membership function than another city with 12 customers. The membership value always lies between 1 and 2.

The second and third factors are based on memory functions embedded within the model. MURRAY contains two types of memory. The simplest type, on which the second weighting factor is based, is the model's ability to remember previously issued commands. MURRAY is restricted from reissuing the same command within a certain span of time. The factor derived from this kind of memory is a function of time and frequency of usage of a given command. Commands that are used more frequently are more "automatic", and thus, are retained for less time in memory [Henneman and Rouse 1984c]. The numerical value resulting from this factor is also presented to the prioritization of a given rule and always lies between 0 and 1. The more recently a command was issued, the lower is the value of this second factor. (A value of 1 represents the fact that memory retrieval for a given command failed.)

The second type of memory, which is more complicated than the simple command memory described above, is the contextual knowledge that an operator accrues over time due to learning. This type of memory is the basis for the third rule weighting factor. The factor is bounded between 1 and 2 and its value increases with contextual representativeness. This form of memory is implemented by the context frame structures that were explained earlier and accessed by the inference mechanism upon application of rules that allow actions to be activated from the contextual memory instead of solely from a system state. Information in the context frames is updated whenever a particular city is displayed. If no updates occur, the retention of the context frame decreases over time until it is eventually deleted from the contextual



memory.

The dynamic importance of a rule in the given context is finally obtained by multiplying the a priori weight value of that rule by each of these three factors. The rule that is eventually chosen is determined by ordering the applicable rules in descending order of derived importance in a priority queue and using a head-of-the-line queueing discipline. The first element of the priority queue is the model's first choice of the next CAIN command. The following section presents an example of how this mechanism works.

#### 4. Example

Consider a situation in which the currently displayed cluster is at the highest level of the system. The previously observed cluster was on the second system level below Chicago, and the operator's last command that was issued was an 'up' command. Information available in the cluster frame structure includes the queue sizes of the 16 top level cities and their locations. For simplification, only three cities with the largest queue sizes will be considered (Seattle(17), Chicago(14) and Atlanta(15)). Previously, the cluster below Boston had a large number of customers and Dubuque (in the level below Chicago) indicated a high queue size. This information was retrieved from the model's contextual memory. Also, the cluster below Seattle was recently displayed. Both the 'monitor' and 'test' commands were issued recently. The watch list of cities with recurring failures was also observed fairly recently and the only city on it was Houston.

The rules applicable to this situation (as derived by the Recognition/Classification component of the model) are listed in Table 3. The numbers in brackets represent the importance weight, the factors derived from

the fuzzy functions (fuzz), from the simple command memory (cmd), and from the contextual memory (ctxt), and finally the overall dynamic importance of the rule is listed. All the values of the three weighting factors can be explained by the above given information about the situation. For example, the dynamic importance for 'down (Chicago)' is obtained by its current queue size of 12 (1.20), the fact that the same command was given before (0.60) and its contextual situation (1.10).

During the prioritization process, 'down (Seattle)' is initially selected as a command (\*). The next several possibilities are not considered as commands since their final importance is less than 72. The 'test' command ultimately yields the highest priority in the given context (\*\*), and thus, is implemented as the next CAIN command. It is interesting to see how close some of the prioritization decisions are. This phenomenon (which suggests that in a given situation more than one action may be 'correct') will be further discussed in the next section when validation issues of the model are addressed.

---

	<u>Command</u>		<u>Weighting Factors</u>			<u>Final Weight</u>	
			fuzz	cmd	ctxt		
#2	down(Seattle)	[47	1.70	0.90	1.00	72]	(*)
#2	down(Chicago)	[47	1.20	0.60	1.10	37]	
#2	down(Atlanta)	[47	1.50	1.00	1.00	70]	
#3	mon	[36	1.00	0.95	1.00	34]	
#6	test	[84	1.00	0.87	1.00	74]	(**)
#11	mon	[44	1.00	0.95	1.00	42]	
#20	down(Boston)	[25	1.10	1.00	1.30	36]	
#21	down(Dubuque)	[24	1.05	1.00	1.20	30]	
#22	down(Houston)	[21	1.12	1.00	1.15	27]	

---

Table 3. A Fuzzy Set of Rules

---

## B. Experiment Three: Model Evaluation

### 1. Method

Experimental data were collected to validate MURRAY. Ten junior and senior Industrial Engineering majors participated in a total of nine sessions (3 training, 6 data collection) monitoring and controlling CAIN. Subjects read 2 sets of written instructions that described CAIN and its operation prior to Sessions 1 and 2. At the end of Session 3, subjects took a quiz to verify their knowledge of CAIN and to assess their level of contextual knowledge. Subjects took a similar quiz at the end of Session 9. Each session lasted approximately 45 minutes, and subjects were paid \$50.00 for their participation.

Independent variables considered in the experiment were Session (6 levels) and Subject (10 levels). Session was of interest to assess if subject or model performance (and level of agreement between the two) improved or degraded with time. Individual subject performance was of interest to assess if degree of model-subject agreement was a function of individual strategy differences.

Comparison of MURRAY and subject performance was done in two ways. First, an "open-loop" comparison was made in which subject performance was compared with MURRAY's performance. Second, a "closed-loop" analysis was performed. Subject data files were replayed concurrently with a version of MURRAY. Whenever a subject action was performed, MURRAY generated the action it would implement, along with a list of its other applicable rules. The subject's action was then implemented. This form of analysis allowed an action-by-action (or process) performance comparison to be made.

## 2. Open-loop Evaluation

MURRAY's performance was compared to subjects' performance in a number of ways and, from all perspectives, MURRAY consistently performed very well. For example, MURRAY's performance on such measures as mean customer sojourn time, number of customers served, and fraction of failures repaired was always between the best and worst subject's performance and usually better than average. When the experimental results were averaged across sessions for each of the subjects, MURRAY outperformed all of the subjects. The only measure for which this result did not hold was the fraction of failures found: MURRAY repaired a smaller fraction of failures than most subjects. This result follows, however, from the fact that MURRAY allowed fewer failures to occur; in short, MURRAY's control resulted in a more stable system. There was not, however, any statistical difference between MURRAY's and subjects' performance as measured by Duncan's Multiple Range Test.

Subject and MURRAY performance was also compared based on individual command usage. A comparison of single command usage showed a high degree of similarity; major differences involved MURRAY's preference for monitoring the system. This activity resulted in much information being displayed. Subjects were apparently reluctant to ask for all of this information, whereas MURRAY could easily process all of these data. A comparison of command sequences showed similar results: MURRAY tended to favor commands that would generate the most information on which to base future actions.

## 3. Closed-Loop Evaluation

Subject and MURRAY performance were also compared on an action-by-action basis, thereby facilitating a comparison of subject and model in exactly the

same environmental conditions. This type of analysis allows a validation of the behavioral processes and representations present in the model. Matches were differentiated in three different ways: Type I - same command issued by MURRAY and subject at same time; Type II - same command, different time (i.e., one command earlier or later); Type III - commands that belong to the same class, issued at the same time (e.g., a 'down' command to different clusters). In addition, subject commands were compared to the first three choices that MURRAY had listed in its priority queue of applicable actions. By averaging across subjects and sessions, MURRAY matched subjects' actions across Type I, II, and III matches 75.3% of the time. MURRAY exactly matched (Type I matching) subject performance 64% of the time. These results are impressive given the subjective nature of the rule identification method and development of prioritization weights.

These results for each experimental session were analyzed with ANOVA. The difference between sessions was not significant, whereas the difference between subjects was highly significant ( $p < 0.0001$ ). This result suggests that some subjects used different strategies from the model but did not change them over time.

It can be argued that a critical decision that subjects must make in this task is when a new cluster should be displayed and which one it should be. One reason for the 25% of the commands that were not explained by the model was found by comparing the degree of matches with just the 'down' (or 'change screen') commands. Considering only the first three of MURRAY's choices, subject performance (with respect to only 'd' commands) agreed just 12% of the time (Type I matching). The differences between subjects were significant ( $p < 0.0001$ ), but there were no significant differences between sessions. This

result is consistent with the findings of Henneman and Rouse [1986]: humans not only use symptomatic and topographic search strategies, but also use serendipitous and other random-appearing search strategies; these strategies are not represented or supported by MURRAY in its task knowledge. Nevertheless, relying on its task description provided in the rule base, MURRAY resulted in uniformly excellent performance. Therefore, a model-based aid might be useful in providing the operator with procedural instructions; MURRAY could support the operator with additional or alternative strategies to monitor or control CAIN. In addition, MURRAY could provide support in accessing the network by identifying problem areas that are most critical.

### C. Conclusions

To summarize, MURRAY proved to be a reasonable means of describing human behavior in a complex monitoring and control task. Open-loop analysis of model performance indicated that the model consistently did as well as human operators. Closed-loop, action-by-action comparison of subject and MURRAY performance revealed a high degree of behavioral congruence. Thus, it appears that the structures and mechanisms present in the model produce quite similar behaviors to humans' structures and mechanisms used in performing this task.

Nevertheless, it should be noted that the level of matching was not perfect. Both MURRAY and human operators appear to have different strengths that are useful in this environment: MURRAY is good at prioritizing tasks; the human operator is good at improvising flexible search strategies. Thus, a combination of the two could result in improved overall system performance. The next step in this research program, therefore, was to implement a human performance aid based on MURRAY. Such an aid should provide cognitively plausible assistance to the human operator.

#### IV. AIDING HUMAN PERFORMANCE

Aiding human performance in a system may be done in many ways. For example, it may be possible to aid human performance simply by altering the characteristics of the display of information to the human operator [Mitchell and Saisi 1986]. Alternatively, the aid may provide advice to the human based on some normative representation of a task, such as multi-attribute utility theory [Freedy, et al. 1985]. Still other approaches may use system simulation to allow the human operator to ask "what if" questions of potential actions [Yoon and Hammer 1986]. Coupled with decisions regarding the selection of an appropriate aiding scheme are decisions concerning task allocation. For example, if an aid is able to suggest appropriate operator actions, it might be acceptable to allow the aid to implement its own suggestions in some situations.

In the context of CAIN, one can imagine potential operator performance aids. A simple alteration of the displays (e.g., highlighting the most salient visual cues) could likely lead to a performance improvement. Another approach might be based on the complexity measures described earlier: the aid could make recommendations based on actions that would reduce complexity by the greatest amount. In this section, one particular approach to aiding the human operator is developed and evaluated. The approach proposed here is based on the model of human operator performance, MURRAY, that was discussed in the preceding section. Since the model contains knowledge structures and mechanisms congruent with those underlying humans' behaviors, the model should be effective in providing meaningful advice to the human operator [Knaeuper and Morris 1984]. Thus, the model-based aid evaluated here is significantly different from the decision support available from expert systems or other

normative approaches. Although MURRAY's advice is always derived from a set of if-then rules (as is an expert system), MURRAY's decisions are based on its embedded knowledge structures (i.e., contextual and system) and its prioritization mechanism to resolve conflicts among rules. The model is only expert in the sense that it makes use of all available information on the complex CAIN display, has a good memory, is not pressured by time-critical situations, etc.

The implementation of the MURRAY-based aid is largely one of designing an appropriate interface. The design of this interface is critical in that the operator should be neither overloaded with information nor preoccupied with requesting advice. In view of the complexity of the existing CAIN display and associated operator functions, the decision was made to implement a simple, straightforward interface for the aid. The mechanism works as follows. MURRAY operates in real-time in parallel with the human operator who is controlling the system. MURRAY suggests a single command to the operator upon request, i.e., whenever the operator issues an 'h'-command ('help'). MURRAY's highest ranked choice for the next command is presented on the CAIN display next to the command entry line at the lower center part of the display [Figure 2]. Considering factors such as the operator's mental workload and the time-constrained dynamic environment, this rather simple augmentation of the display was selected over other possible implementations, such as multiple command options, displaying further information relative to MURRAY's prioritization process, or even adding another display with aiding information. This interface is directed at the expert end user (such as the CAIN operator) rather than a sporadic novice user.



## A. Experiment Four: Aid Evaluation

### 1. Method

The evaluation of the on-line aid was performed by augmenting Experiment Three described in the previous section. The main goal of Experiment Four was to assess the effects of on-line aiding on operator performance in the CAIN environment. The experimental design used to evaluate this issue was a between-subjects design in each of two treatment groups: unaided (using the subject performance data from Experiment Three) and aided operation of CAIN (a new group of 10 subjects). Thus, the treatment structure is a one-way factorial design with aiding being the independent variable of interest.

A second group of ten paid subjects participated in operating CAIN for nine sessions. Instructions, training, and questionnaires were presented in three sessions as in Experiment Three. The difference in this experimental condition was the availability of the on-line aid. The instructional material was augmented by a description of the 'help' command. The new command was introduced in the second training session. The subjects were instructed to use the aid when uncertain about what to do next or to enhance their own strategies. Subjects were also told to implement the aid's suggestion only if they felt it was reasonable.

### 2. Results

From several perspectives, the aid had no impact on subject performance. ANOVA revealed no statistical differences between groups on the various measures of subject performance, although the aided group frequently performed slightly better. A comparison of command usage also showed no major systematic differences between groups.

At first observation, these results are disappointing. However, a more fine-grained analysis of the data revealed ways in which the aid was quite helpful. First, although there were no statistical differences between groups, aided subjects were able to find failures faster than unaided subjects, thus maintaining a more stable system. Accordingly, unaided subjects had more failures occur during their experimental sessions. Second, the aid enjoyed a high level of acceptance by subjects. On the average, 83% of all commands that were suggested by the aid were actually implemented by subjects. Given that aid requests constituted only 8% of all commands issued, however, this high level of acceptance was not reflected in the overall performance scores. These results suggest that more emphasis should be given in the future to training operators in the use of the aid to illustrate its benefits. Third, the questionnaires completed by subjects at the end of the experimental sessions indicated that aided subjects had a higher level of contextual knowledge (as measured by number of second-level city locations correctly recalled) than unaided subjects.

Finally, as emphasized in Section III, one of the strengths of MURRAY is its ability to prioritize tasks. In fact, a key to good performance in this task is the ability to decide which part of the network should be observed next. It is interesting to note, therefore, that the percentage of times a 'd'-command suggested by the aid led to finding a failure was 34%; the percentage of times any 'd'-command issued by a subject that was not suggested by the aid led to finding a failure was only 10%. Clearly, from this perspective the aid was quite beneficial in providing useful aid to the operator. Nevertheless, since the aid was requested infrequently, these fine-grained results were not reflected in the overall performance scores. As mentioned

above, the low level of use masked any overall performance improvement.

## B. Conclusions

In the final phase of this research program, an on-line performance aid based on MURRAY for human monitoring and control in a large-scale system was introduced, described, and evaluated. Model-based on-line aiding was selected because previous efforts have shown its potential benefits [Knaeuper and Morris 1985]. Experimental results, however, failed to show significantly improved overall performance of aided subjects. Nevertheless, more fine-grained evaluation of the results demonstrated subtle subject improvement in some performance aspects. One of these aspects was a more stable operation of the CAIN system by aided subjects. The second and most important result was improved subject performance in the critical decision of selecting which part of the network to observe next. The aid provided clear performance improvement with respect to failure-detection strategies. These subtle performance improvements suggest that further research is needed to determine if alternative implementations of the aiding approach could result in more definitive results.

Several other aiding approaches are viable given the experimental results presented in this report. For example, the fact that subjects did not request the aid very often suggests that different results could be obtained if the model's suggestion was always available. (Such an approach would be consistent with the model-based aid used by Knaeuper and Morris [1984]). Alternately, the aid could present its recommendation only if the derived importance ranking of the rule exceeded some threshold value. A related approach would be to emphasize the use of the aid through training as mentioned above. Another alternative would be to alter the strategy of the model so that it

---

would support problem-solving strategies significantly different from the human's. Zinser [1986], for example, found that if the a priori weightings of the model's rules that were related to contextual knowledge were increased, command matches with human performance decreased; the model began to place more emphasis on context-dependent strategies (e.g., recalling that Dubuque has recurring failures). Given that approximately half the failures in CAIN were dependent on the context, a strategy based more on contextual recall and augmented with 'normal' human strategies should be very effective. In light of the ambiguity of the current results, these ideas merit careful further consideration.

## V. CONCLUSIONS

The research described in this report has considered human performance in the monitoring and control of large-scale systems from many perspectives. Initial efforts empirically examined the effects of system design parameters on human performance. The results clearly illustrated the problems that people have in controlling a multiple-level system. Large performance differences were noted when the number of system levels increased from two to three. Multiple-level systems tend to mask failure symptoms from the human operator. Although such systems protect upper system levels from the effects of failures, when the failures do propagate upwards, their effects are more serious. Unfortunately, the results presented here indicate that people tend to wait until symptoms emerge rather than pursuing a more active search strategy.

Similar comments can be made regarding the degree of redundancy present in a system. Increased connectivity between system parts led to improved automatic system performance but degraded human failure-diagnosis performance. Thus, system designers need to be aware of the tradeoffs that can be made between supporting automated system control and human failure-diagnosis performance. Moreover, if the physical structure of the system cannot be altered to support good human performance, then aids must be designed within the system to cause the human operator to adopt effective control strategies.

Other human limitations in dealing with large multiple-level systems were also noted. For example, when contextual information was introduced to the system, humans did not adopt strategies that took any great advantage of this information, even though they were aware of certain types of failure that could be located more readily by using contextual knowledge. Humans used a rather mechanical strategy that did not rely on the context. People also had

---

difficulty in prioritizing subtasks in time-critical situations. It was shown that by relying on a model-based aid with a very good prioritization method the human could make better search decisions. Again, people tended to learn one way of performing the task and not change as the environmental conditions shifted.

The notion of relying on an aid based on human cognitive functions deserves much closer scrutiny. Despite some ambiguity, the results discussed in this report are promising: subtle performance improvements were shown when the aid was used by subjects. The approach is consistent with the views espoused by Rasmussen [1985] regarding the support of human operators in complex systems. In particular, Rasmussen argues that an aspect to consider in the design of a system is "a representation of the information processing capabilities and limitations of the decision maker and of the subjective formulation of goals and criteria for choice among possible strategies..." MURRAY provided such a representation of the human operator that was shown to be of use in decision support. MURRAY gave "cognitively plausible" advice to the human operator when that information was needed.

Nevertheless, the CAIN system (as augmented with MURRAY's advice) is limited in the support it can provide the operator. Rasmussen [1985] argues that systems should support an operator at various levels in an abstraction hierarchy of functions and according to various levels of aggregation. Although CAIN does support various levels of aggregation, CAIN's level of abstraction to the human operator is fixed. Future work should concentrate on defining the system functions and representations at various levels of abstraction.

Concurrent efforts should be directed at considering some of the issues related to aiding mentioned in Section IV. In particular, alternate interface

design methods, task allocation strategies, and issues related to user acceptance should be considered. Also, the notion of implementing a model with search strategies complementary to (as opposed to coincident with) human strategies (e.g., based on the contextual information) should be explored. Aids based on models complementary to human strategies may have more potential to improve overall performance but may be difficult for the human operator to understand. On the other hand, aids based on models coincident with human strategies may be easy for the human to understand but may not enable any improvement over unaided performance. This potential trade-off deserves more consideration.

In summary, the material presented here has made several important contributions. First, it has added to a general understanding of the relationship between system design characteristics and human performance. Second, from a theoretical perspective, this project has contributed a framework for measuring the complexity of a system based on the physical system characteristics and the human's understanding of these characteristics. Third, a model of human performance was proposed and evaluated that was made up of several different interacting knowledge structures and cognitive mechanisms. The model was shown to produce behavior consistent with human performance. Finally, this model was shown to be effective as a means of aiding human performance in a complex monitoring and control task.

## REFERENCES

- American Telephone and Telegraph. Connecting a nation. Administrative Training Center, Morristown, NJ.
- American Telephone and Telegraph. Notes on the network: Section II. Administrative Training Center, Morristown, NJ.
- Anderson, J.R. Language, Memory, and Thought. Hillsdale, NJ: Lawrence Erlbaum, 1976.
- Ash, G.R. and V.S. Mummert. AT&T carves new routes in its nationwide network. AT&T Bell Laboratories Record, August 1984, pp. 18-22.
- Box, G.E.P. and G.M. Jenkins. Time series analysis: forecasting and control. Holden-Day: San Francisco, 1976.
- Freedy, A., A. Madni, and M. Samet. Adaptive user models: methodology and applications in man-computer systems. In W.B. Rouse (Ed.), Advances in Man-Machine Systems Research (Vol. II). Greenwich, CT: JAI Press, Inc. 1985.
- Henneman, R.L. Human problem solving in large scale dynamic networks. Ph.D. Dissertation, Georgia Institute of Technology, CMMSR Report No. 85-1, 1985a.
- Henneman, R.L. A model of human performance in a large scale system. Proceedings of the International Conference on Cybernetics and Society, Tucson, AZ, 1985b, pp. 527-531.
- Henneman, R.L., and W.B. Rouse. Human problem solving in large scale networks. Proceedings of the IEEE International Large Scale Systems Symposium, Virginia Beach, VA, 1982, pp. 293-297.
- Henneman, R.L., and W.B. Rouse. Human performance in monitoring and controlling hierarchical large scale systems. Proceedings of the Human Factors Society 27th Annual Meeting, Norfolk, VA, 1983, pp. 685-689.
- Henneman, R.L., and W.B. Rouse. Measures of human performance in fault diagnosis tasks. IEEE Transactions on Systems, Man and Cybernetics, SMC-14, (1):99-112, 1984a.
- Henneman, R.L., and W.B. Rouse. Human performance in monitoring and controlling hierarchical large scale systems. IEEE Transactions on Systems, Man and Cybernetics, SMC-14, (2):184-191, 1984b.
- Henneman, R.L., and W.B. Rouse. Assessing the complexity of a large scale system: measures of system structure and human strategy. Proceedings of the 1984 IEEE International Conference on Systems, Man and Cybernetics, Halifax, NS, 1984c, pp. 68-72.



- Henneman, R.L., and W.B. Rouse. On measuring the complexity of monitoring and controlling large scale systems. IEEE Transactions on Systems, Man and Cybernetics, SMC-16, (2):193-207, 1986.
- Hunt, R.M., and W. B. Rouse. A fuzzy rule based model of human problem solving. IEEE Transactions on Systems, Man and Cybernetics, SMC-16, (2):112-119, 1984.
- Keeney, R.L. and H. Raiffa. Decision making with multiple objectives. New York: Wiley, 1976.
- Knaeuper, A. and W.B. Rouse. A rule based model of human problem solving behavior in dynamic environments. IEEE Transactions on Systems, Man and Cybernetics, SMC-15, (6):708-718, 1985.
- Knaeuper, A. and N.M. Morris. A model based approach for on-line aiding and training in process control. Proceedings of the 1984 IEEE International Conference on Systems, Man and Cybernetics, Halifax, NS, 1984, pp. 173-177.
- Minsky, M. A framework for representing knowledge. In P. Winston (Ed.) The Psychology of Computer Vision. New York: McGraw Hill, 1975.
- Mitchell, C.M. and D.L. Saisi. Use of model-based qualitative icons and adaptive windows in workstations for supervisory control systems. Submitted for publication, 1986.
- Mocenigo, J.M. and D.M. Tow. Managing a network that won't sit still. AT&T Bell Laboratories Record, August 1984, pp. 23-26.
- Nawrocki, L.H. Computer-based maintenance training in the military. In J. Rasmussen and W.B. Rouse (Eds.) Human Detection and Diagnosis of System Failures. New York: Plenum Press, 1981.
- Newell, A. and Simon, H.A. Human Problem Solving. Englewood Cliffs, NJ: Prentice Hall, 1972.
- Paap, K.R. and R.J. Roske-Hofstrand. The optimal number of menu options per panel. Human Factors, 28, (4):377-385, 1986.
- Rasmussen, J. The role of hierarchical knowledge representation in decision making and system management. IEEE Transactions on Systems, Man and Cybernetics, SMC-15, (2): 234-243, 1985.
- Rouse, W.B. Models of human problem solving: detection, diagnosis and compensation for system failures. Automatica, 19, (6): 613-625, 1983.
- U.S. Army. Combat communications within the division. Field Manual 11-50, March 1977.
- U.S. Army. Communications-electronics management system. Field Manual 24-22, June 1977.

- Viteri, E. A rule based model of a human operator in a complex communication network. M.S.I.E. Thesis, Georgia Institute of Technology, CMMSR report no. 84-3, 1984.
- Wickens, C.D. Engineering Psychology and Human Performance. Columbus, OH: C.E. Merrill Publishing Company, 1984.
- Yoon, W. and J.M. Hammer. Aiding the operator during novel fault diagnosis. Submitted for publication, 1986.
- Zinser, K. Modeling and aiding human decision making in operating and supervising dynamic systems. M.S.I.E. Thesis, Georgia Institute of Technology, CMMSR report no. 86-4.
- Zinser, K. and R.L. Henneman. Evaluation of a model of human performance in a large scale system. Proceedings of the 1986 IEEE International Conference on Systems, Man and Cybernetics, 1986, pp. 869-874.